A Study of a Health Enterprise Information System, Part 4 - Conceptual Data Modeling

by
Jon Patrick
School of IT
The University of Sydney
Australia

Abstract

A set of 6 entity-relationship diagrams representing subsystems of the Cerner Millennium products are analysed for design strengths and weaknesses. The results demonstrate that about 25% of the structures in the diagrams have points of weakness that could lead to errors in database design and programming errors. They also effect the correct storage and retrieval of patient data in a working clinical information system. There is one seemingly serious design weakness where two different important entities, PERSON and PRSNL are given the same primary key, person_id, in the diagrams, yet a putative primary key attribute for PRSNL, prsnl_id, can found throughout the diagrams in foreign key roles and as a compound in other attribute names. The consequences of these weaknesses such as failure to adhere to Entity Integrity and Referential integrity rules are explored.

1. Introduction

This essay is a review of six Entity-Relationship Diagrams representing the conceptual data model used in some of the Cerner Millenium products. An analysis of these diagrams has been undertaken to assess their conformance with good modeling practice. The detailed list of identified weaknesses is presented in Appendix 1. A review of a data model needs to consider a number of factors, such as:

1. Consistency with the original requirements gathered,
2. Internal consistency,
3. Usefulness for maintenance of an implemented system.
4. The role as a document of agreement between the vendor and client as a representation of requirements.

In this study we do not have the original systems requirements but in some circumstances we can use general knowledge of the clinical context to make a few observations. The schema diagrams made available are of a size and range across 6 subsystems so that some reasonable assessments of internal consistency can be assessed. As the ERD diagrams are a primary record of design for any information system then these diagrams can be used to address some of the issues they might introduce should they be used for development and maintenance of both the design and software implementation.

In studying these observations presented in this essay a number of caveats must be considered. Firstly, software weaknesses do not necessarily constitute errors. An error can usually only be defined by matching the users specified requirements with the models of their worlds. However in the case of the Cerner software the users rarely if ever get the opportunity to do this matching. Rather the models are delivered to the users “canned”, that is, they are embodied in the delivered software with minimal transparency that allows them to be reviewed. Secondly, it is possible that there is a discrepancy between the design and the implementation model and that the changes introduced in the build phase of development have not been fed back to the design documents. That said, it is important that such feedback be appropriately applied in a large organisation as the documentation serves three important functions: being the basis for developing the system; subsequently for maintaining the system; and, the starting point for new staff to be trained in design of the system.

2. Terminology and Important Design Criteria

This section is presented here to aid the reader who may not be experienced in the terminology or design of data models and relational database schema. It is our goal to lay out in more detail the important criteria we have applied in this assessment. Issues of importance addressed herein are:

1. The naming conventions used in a data model
2. The declaration of Primary Keys
3. The use of Primary Keys as Foreign Keys
4. The use of 1:M relationships between entities
5. The meaning of entities that are unconnected to other entities.

The definition and importance of some of these terms are now presented. An early step in designing a data model an early step is the identification of the entities about which data will be collected. In a clinical context there are obvious entities such as patients, staff, medicines, wards, and departments. But there are also less obvious entities such as diagnoses, and medical procedures and events like admissions and discharges. There are two important aspects about creating entities in a data model. They must each have an unambiguous name that resembles as closely as possible their name in the real world, and they must have a Primary Key (PK) which is the attribute that uniquely identifies each member of the entity class. The wording of the PK often carries a convention that allows it to be rapidly and unambiguously associated with the name of the entity it defines, hence the entity such as PATIENT should be given the PK name *patient_id*. The PK must be unique for each record of information in the Entity storage table, for example each patient in a hospital must have their own unique identifier and the database management system will maintain this uniqueness if it is told which attribute is the PK of an entity. This is a process called Entity Integrity. Although simple in concept it becomes vitally important in the preservation of the uniqueness of the individual patient's record of data when the record is spread across a substantial number of entity tables as happens in a large clinical information system.

The PK of an entity serves an important role when entities need to be decomposed or constructed by some combinational process. When there is information about an entity that is repeated many times, such as the problem list of a patient then another entity has to be created, we will call this the PATIENT_PROBLEM_LIST entity. The relationship between the PATIENT entity and the PATIENT_PROBLEM_LIST is known as a one-to-many relationship, as for every one patient there are many problems, and is typically written in the shorthand style of "1:M". The PATIENT_PROBLEM_LIST entity requires a PK made up of the PK of the PATIENT entity and an identifier for each problem of a given patient. This can be called a counter and given the attribute name *patient_problem_list_counter*. The unique identifier for each problem of each patient is the combination of the two attributes *patient_id* and *patient_problem_list_counter*. The use of the *patient_id* in the PATIENT_PROBLEM_LIST entity is known as a Foreign Key (FK). It serves an important role of preventing the problem list items of a patient from being separated from the patient they belong to, as the same problem will be present on the problem list of many patients. The process of maintaining the reference of the PK as a FK is called Referential Integrity, and in database management software it consists of automatically checking that the FK value is a legal value in the set of PKs from which it is copied from. It also ensures that anything done to the PK value is reflected where appropriate in all the FK usages of that value. For example, a patient may come to the Emergency Department of a hospital being unconscious and without any identification, so they are allocated a temporary *patient_id* until they can be identified, at which time all the records especially those that are referenced by FKs have to be changed to the correct PK value for the identified patient. Failure to make that change will cause parts of the patient's record to be dissociated from their true record and float in the system as unlinked and therefore inaccessible content. In such circumstances staff would discover that data was missing and have no way of retrieving it or discovering where it was held in the system.

In inspecting the diagrams some of these matters will be subject to scrutiny to determine if the design might lead to the contravention of the principles of Entity Integrity and Referential Integrity in ways that could lead to loss of data or loss of access to data.

3. The Data Set

There are 6 diagrams in total which are drawn in the style of the Microsoft Visio drawing software usually with the following conventions. The diagrams consist of a set of single columnar boxes, one box per entity, each consisting of a column containing a list of attribute names, their data type and optionally the string "FK" being an abbreviation for "Foreign Key". At the top of the column is a partitioned section where the Primary Key (PK) of the entity is meant to be written. Immediately above the box is a string, written in upper case by convention, which stands for the name of the entity being described by the attribute names in the box. The 1:M relationship between entities is represented by a line joining two boxes and a "crow's foot" symbol at the end of the line which is the "many entity".

There are 6 diagrams available for analysis.

1. Patient Tracking ER Diagram (consisting of 4 pages)
2. DCP Power Chart ER Diagram (consisting of 2 pages)
3. The DCP Task List ER Diagram (consisting of 1 page)
4. Patient Management ER Diagram (consisting of 4 pages)
5. DCP Document Management ER Diagram (consisting of 2 pages)
6. Security Application ER Diagram (consisting of 1 page)

4. The Methodology

The diagrams have been assessed for their conformance to consistency and good design criteria. Initially a manual inspection was made of each of the diagrams to establish the type of information they contained and what potential issues could be identified. This lead to the observation that there were potential consistency issues with the declaration and use of keys throughout the diagrams, hence an analysis was made on the reuse of PKs as FKs across all diagrams. This study in turn led to issues on the conventions for naming of attributes which was subsequently studied. Lastly the use of 1:M linking between entities seemed to be tied into issues of the consistency of naming keys so it was analysed as well.

As a further stage of analysis the Patient Tracking diagram was copied into a spreadsheet to make richer comparisons. The attribute names and entity names were loaded into two columns of a spreadsheet and then sorted alphabetically on attribute name. This readily allowed identification of the reuse of the same attribute names and all the entities they were used in. Hence we could investigate when attributes might be reused and not declared as FKs, thus allowing detection of potential violation of the Referential Integrity rule.

5. Limits of the Study

The diagrams of themselves do not identify the actual implementation characteristics that might cause difficulties for consistencies in data management. However, weaknesses found at the design stage generally indicate the scale of faults that might be found in the implementation. There is always likely to be some gap between documentation such as discussed herein and the actual implementation as problems discovered and corrected on implementation should be fed back into the documentation, but sometimes aren't. Nevertheless, this essay can be considered as a data model review process that has its validity in wishing to identify potential problems with the data model so that they might be designed out of the implementation rather than designed into the implementation where they will be much more costly to discover and remove.

6. Information Extracted from the Diagrams

The range of information about entities, links between entities, PK usage, and putative FKs has been identified from the 6 ERDs (see Table 1). A detailed description of each element counted in Table 1 is recorded in Appendix 1.

2. Analysis of Weaknesses in the ERDs

The diagrams were studied for general issues to get an understanding of their overall content and coherence. Subsequently each entity was studied to identify the principle information about PKs and FKs, the auditing information that appears in almost every table, and the information relevant to individual entities. This process lead to an understanding of the principal weaknesses over all the ERDs. It may be that there are other weaknesses in the designs that we haven't completely recognised but the amount of coverage provided should be sufficient to make acceptable generalisations.

2.1. Principal Weaknesses

An analysis of the detailed documents has lead to building a summary apparatus of 4 Principal Weaknesses (PWs), namely:

1. Inconsistent use of appropriate primary key (PK) names and naming conventions for important entities.
2. The secondary use of PK attributes in entities without declaring them to be Foreign Keys (FKs).
3. 1:Many relationships that do not include the PK of the 1-side entity as a FK in the Many-side entity of the relationship.
4. Missing relationship links not drawn on the diagram.

A compilation of the principal weaknesses is presented in Table 1. It is evident just by inspection that the weaknesses occur more frequently than we would expect by chance. The PW3 occurs much less frequently than the other 3 PWs as it requires 3 entities to be defined. On average there are about 2 PWs for every entity across the 6 ERDS. The worst is the Security Application which has more than 1 PW for each entity, that is 1.2. Even the best performing ERD has 1 PW for each 2.25 entities which still seems remarkably high for professional quality work, and a period of many years to refine the ERDs and get most of the errors cleaned out.

> *The overall analysis shows that there is about 2 weaknesses for every 3 entities over a total of about 350 entities, which is seemingly higher than one would expect in robust industrial strength systems design.*

The details of the weaknesses identified in each entity of each diagram are presented in Appendix 1. In Appendix 2, the original ER diagrams have been re-drawn showing only the content and sufficient context for understanding the issues discussed in this essay. Content in the ERDs not relevant to this discussion has not been reproduced.

Issues	Diagram 1	D2	D3	D4	D5	D6	Total
	Patient Tracking	Power Chart	Task List	Patient M'ment	Doc't M'ment	Security App	
1. Connected entities	50	25	21	57	45	20	218
2. Unconnected Entities	10	10	2	29	13	13	77
3. Total entities	60	35	23	86	58	33	295
4. Links that have matched FKs	45	16	29	57	41	5	193
5. Links that don't have matched FKs	16	2	6	8	1	8	41
6. Total links	61	18	35	65	42	13	234
7. FKs with matching links	45	16	23	58	35	5	182
8. FKs without matching links	81	2	6	34	37	8	168
9. Total FKs	126	18	29	92	72	14	350
10. Entities with no PK	2	2	0	4	3	2	13
11. Names ending in "-id" but not declared an FK	41	22	18	55	25	22	183

Table 1. Summary of items extracted from each of six ERDs of the Cerner Millenium product range. See Appendix 1 for itemised list of putative weaknesses.

2.2. Some Issues on Naming Conventions

2.2.1 Use of "id" in Attribute Names

The string "id" plays a significant role in the naming of entities. The predominant manner of forming PK names across the 6 diagrams is to use the name of the entity and append the string "_id" to the end of it. This is a very good practice as it makes it easy for the modeler and programmer to immediately recognise the association between the entity and the attribute name. In many parts of the diagrams this method is used effectively and consistently to make the diagram easily readable. Unfortunately there are enough examples where this systematic approach is abandoned leading to serious weaknesses in the design, and therefore one's ability to interpret its meaning.

There are 5 entities used extensively throughout the 6 diagrams: ENCOUNTER, ORGANISATION, LONG_TEXT, PERSON and PRSNL. The usage of the first three of these entities do not show any behaviours that indicate any weakness in design, however the usage of the latter two raises serious concerns.

2.2.2. Use of "cd" in Attribute Names

In a number of places the suffix string "_cd" is affixed to names of FKs. The meaning of this suffix is uncertain but is possibly an abbreviation for "code". This is a weakness of design as it leaves uncertainty as to the type of information the attribute is meant to carry. Code numbers are certainly a class of information that one might reasonably expect to be used as PKs but in the case of the 6 diagrams there is no linking to entities that enables one to assess the potential meaning of the PK and therefore the validity of the FK. This is a

design weakness but no in depth study has been made of the usage of cd-attribute names due to the lack of suitable information in the diagrams.

ERD	PW1	PW2	PW3	PW4	Total PWs	Total Entities in ERD	Proportion PW/ Entity
TASK LIST	3	4	2	5	14	23	0.61
DOCUMENT MANAGEMENT	7	8	1	8	24	58	0.41
PATIENT MANAGEMENT	12	27	1	22	62	86	0.72
Patient Tracking	3	21	7	11	42	60	0.70
POWER CHART	4	7	0	9	20	35	0.57
SECURITY APPLICATION	16	11	2	10	39	33	1.18
TOTAL	**45**	**77**	**12**	**65**	**199**	**295**	**0.68**

Table 2. The frequencies of the principal weaknesses in the Entity-Relationship Diagrams.

2.2.3 Differences between PERSON and PRSNL Entities

A configuration that was apparently problematic for the designers throughout the design of all the ERDs, was the differentiation between the PERSON and PRSNL entities. Both entities were provided with the same PK name *person_id*. The attribute name *person_id* naturally fits as the PK attribute of PERSON. One can see the attribute *prsnl_id* used often in the schema both as a PK, FK and in the composition of attribute names, yet the PRSNL entity is never linked to those entities. This is counter-intuitive to good design where the PK of the PRSNL entity should have been *prsnl-id*. The problems arising from this design are twofold. Firstly, the two entities are apparently different, the first being the reference to patients and the latter to staff in the institution. Giving them a common PK is not optimal. Secondly, where a modeler uses the attribute *person_id* in a design it is not evident as to which entity they are intending to use. This danger has clearly been perceived by some modelers as they have changed to using references to *prsnl_id* to resolve the ambiguity. Unfortunately that does not solve the problem for staff who take up the development and maintenance tasks at a later time as they will then see an attribute that appears to have no primary location at which it is defined.

The use of the same attribute name as the PK for two different entities is especially confusing as it doesn't enable one to know which entity is really meant to be referenced when *person_id* is used as a FK. Also it doesn't enable one to interpret the true entity that is being linked too when the attribute *prsnl_id* is used. Our software quality advisors believe that the PERSON entity is intended to represent the patient and the PRSNL is intended to represent the user, although this is not entirely hard and fast from studying the data in the entities. This ambiguity might have been resolved if PRSNL had not been contracted but written in its full form, say PERSONNEL, if that is what is meant, but we cannot be sure this is the true meaning of the entity. Further confusion on this point has been created by the introduction of attribute *user_id* in some entities.

2.3 Overview of ER Diagram Analysis

2.3.1. Unconnected Entities

It is not entirely unusual for some entities to not be connected to any other entities. This type of entity is used to hold background knowledge or look up codes for certain processing tasks. About 25% of entities are not connected to any other entity. However many of these have id-attributes that one would expect to be linked, for example the APPLICATION_INI entity in the Security Application diagram has no PK yet has the attribute *person_id* clearly a key attribute in the whole system yet it is not defined here as a FK.

2.3.2. Links

One would expect every link shown on the diagrams to have the PK at the One end of the link to be inserted into the entity at the Many end of the link. Yet in more than 15% of the links this has not been done. This is a violation of the Referential Integrity Rule.

2.3.3. Foreign Keys

Foreign Keys indicate that the entity is defined by its relationship with another entity. FKs that do not show links to another entity do not enable the reader nor programmer to determine the entity that has to be linked to. This prevents the incorporation of referential integrity into the working system. Approximately 50% of the FKs do not show the necessary links in the diagram. It is of course possible that the diagram has been optimised to omit entities whose links would be self-evident by the naming convention, but such a manoeuvre requires complete confidence in the reliability and use of the naming convention, which of itself is questioned elsewhere in this essay.

2.3.4. Missing Primary Keys

About 5% of entities have no PK. All but two of them have id-denotated attributes that might be used as a PK. The lack of a PK means these entities do not have a unique identifier for each record and therefore the basis of storing and retrieving data from them is uncertain. There can be a legitimate design of an entity without a PK where they are meant to carry a simple set of information, such as a list of cities or geographical districts, but such a construct has to be used with great caution. For example if the members of the list have attributional information that changes over time then this is an unsuitable design strategy.

2.3.5. Non-Foreign Key id-denotated Attributes

These are attributes in entities that end with the string "_id" and therefore by implication are the PK of some other entity in the system. It is clear that two attributes which fall into this definition were used for auditing purposes in a very large number of entities so they were not counted in these totals. Compared to the list of declared FKs the id-denotated attributes are about 50% of the set (183/350). This is a mystifying aspect of the design. It is clear that many of the id-denotated attributes are real PKs in entities shown in the 6 diagrams. Others are most likely PKs from parts of the system outside the set of 6 diagrams. The concern about using PKs and not declaring them to be FKs is the possibility that referential integrity will not be maintained across the entities and therefore potentially either erroneous information will be loaded into patient records, or data will become dissociated from any record or associated with incorrect records so patients will have the contents of someone else's episodes loaded into their record.

2.3.6. Misunderstood Primary Design

There is evidence of a lack of detailed understanding of the overall system in some of the design documents, especially the Security Application ERD. In this document some of the standard entities of the larger system are very obviously mis-labeled with the wrong primary key e.g PERSON with *reference_task_id*, ENCOUNTER with *pathway_catalog_id*, and ORGANISATION with *encntr_id*. The self evidently correct PKs of *person_id*, *enctr_id* and *org_id* can seen elsewhere in the design documents. Such mistakes are disturbing from the greater viewpoint of the whole system as it indicates someone of limited familiarity with the larger enterprise is designing new sub-systems without the requisite knowledge.

3. Conclusions

The ERDs carry a level of weaknesses that is surprising for professional standard CIS. One particular ERD carries the signature mark of very inexperienced staff as not only does it have more than one principal weakness per entity but also there is a number of the major entities in the whole system with mislabeled PKs.

Weaknesses in ERDs do not of themselves define major deficits in the operational system but they do introduce three issues of importance, namely:

1. How new staff to the software production and maintenance team learn enough correct information about the system to execute their development and maintenance tasks correctly,
2. What software design quality strategies are in place to ensure weaknesses in the ERDs are identified and rectified before they get into the operational system.
3. What feedback mechanisms are in place to correct discovered errors in the ERDs so that they are correct for their intended audience and their role in contractual specifications.

The process of determining the optimal design of a significant software engineering task has been established for many years and is well understood. The design process is normally a lengthy and precise activity as it has been established that mistakes that can be identified in design are only one-tenth the cost of

rectification than after development, and one-tenth again if discovered after delivery to the client. For this reason it is expected that design documents would show a significant level of detail, accuracy and consistency for all stages of development.

These diagrams show many examples of missing and ill-formed PK names, undeclared FKs, FKs missing from relationships, missing links, and ill-formed naming strategies for attributes. Such weaknesses do not damn the whole system but they sensitise the analyst and engineer to potentially significant obstacles to creating a clean and reliable implementation. For example absent or incorrectly named PKs makes it a challenge to ensure new records created for storage in tables can be effectively retrieved in future. If records can be coded with the same primary key then it is uncertain which records will be retrieved in the system and if the duplicates will cause one part of a record to never be discoverable or if at unpredictable times different versions of the record might appear in different contexts. Hence the engineer is forced into developing their own strategy for checking the validity of PKs.

The subsequent task to studying the ERDs is to identify the extent to which these weaknesses permeate the implementation in the schema designs and the data stored in the tables themselves. If some of the weaknesses presented in this essay are present in the implementation then it is of greater concern as it will show that there is only a limited strategy by the engineers to detect and eliminate weaknesses. If the weaknesses disappear then one can deduce that the engineering process is performing well but the feedback procedures to the design modelers is less than desirable.

Appendix 1 - Detailed analysis of Weaknesses in the Entity Relationship Diagrams (ERDs) of Cerner Millenium

Note: Entity names are presented in upper case format, like LONG_TEXT, whilst attribute names are presented in lower case italics, like *long_text_id.* Where the issue is one of PK/FKs assignment a question mark has been placed against the attribute in the diagram itself which indicates that the attribute should putatively be marked as FK.

The DCP Task List ER Diagram

Dated 11th August 2006. Total 23 entities

LONG_TEXT: This entity shares a 1:M relation with MESSAGING_AUDIT but the latter does not have the former's PK, *long_text_id* as a FK, although there is a FK attribute called *msg_text_id*. If this is not the appropriate attribute the design violates the referential constraint. (PW3)

MESSAGING_AUDIT: This entity has an attribute *msg_txt_id*. This attribute is also present in TASK_ACTIVITY_ASSIGNMENT and TASK_ACTIVITY without relationship links shown between them. It is perhaps a FK but the original entity to which it belongs to as a PK is not present. (PW2, PW4)

MESSAGING_NOTIFY: This entity contains the attributes, *notify_prsnl_id* and *assign_prsnl_id*, it indicates that the PK of PRSNL should be *prsnl_id* instead of *person_id* as is mentioned. Likewise *assign_prsnl_group_id* and *notify_prsnl_group_id* appear to be FKs to PRSNL_GROUP. (PW1)

ORDER_CATALOG: This entity shares PK, *catalog_cd*, with ORDER_TASK_SYNONYM_XREF. There is a missing relationship link to ORDER_TASK_PROFILE_XREF considering the latter has the attribute *catalog_cd* as a FK. (PW4)

ORDER_TASK: *Reference_task_id* is in the PK for this entity and the entity ORDER_TASK_SYNONYM_XREF, but there is a missing relationship link between ORDER_TASK and ORDER_TASK_SYNONYM_XREF.

ORDER_TASK is linked twice to ORDER_TASK_RESPONSE via the FKs *reference_task_id* and *response_task_id*. It follows that the PK for ORDER_TASK should have been *order_task_id* and the FKs in ORDER_TASK_RESPONSE should have been *reference_order_task_id* and *response_order_task_id* to make them semantically consistent. (PW1, PW4)

ORDER_TASK_SYNONYM_XREF: This entity shares *catalog_cd* with ORDER_CATALOG, *reference_task_id* (as part of the PK) with ORDER_TASK and *synonym_id* with ORDER_CATALOG_SYNONYM of "Power Chart" ERD. However, these attributes are not marked as FKs. It is thereby unclear which is the original entity for these attributes. It also follows that this entity must then have relationship links to the above entities whereas it is shown as a disconnected entity in the schema. (PW2, PW4)

PROFILE_TASK_R: PROFILE_TASK_R and ORDER_TASK_XREF are linked through ORDER_TASK_PROFILE_XREF in 1:M relationships, but the matching attribute in PROFILE_TASK_R, *catalog_cd,* taken from the PK of ORDER_CATALOG, is not declared as a FK. (PW2)

PRSNL_GROUP: The entity PRSNL_GROUP has the PK, *prsnl_group_id,* with a 1:M relationship with the entity TASK_ACTIVITY which does not contain a reference to this attribute. (PW3)

TASK_ACTION: TASK_ACTION uses the PK, *task_id,* of TASK_ACTIVITY as part of a concatenated key, however, it is not declared as a FK. TASK_ACTION is disconnected in the schema but should have a relationship link to TASK_ACTIVITY via the above attribute. (PW2, PW4)

TASK_ACTIVITY_ASSIGNMENT: The putative FKs, *mesg_text_id*, *proxy_prsnal_id* are not declared. This entity has both the entities PERSON and PRSNL in relationship links to it, however both have the PK, *person_id*. There are attributes *assign_person_id* and *assign_prsnl_id* in this entity. This means that there is a matching FK for the

link of the entity PRSNL after using *assign_person_id* for the link to the entity PERSON. It also suggests as evidenced elsewhere the correct PK of PRSNL is intended to be *prsnl_id* but has been incorrectly entered in this and other diagrams as *person_id*. There is a similar problem with FKs as for the entity MESSAGING_NOTIFY. (PW2)

DCP Document Management ER Diagram

Dated 6th Nov 2006. Total 58 entities.

DCP_ACTIVITY_LOG: There appears to be a missing relationship link and FK declaration with PRSNL entity as this entity has an attribute of *prsnl_id*. It shows no relationship links with any other entity. (PW2,PW4)

DCP_FORMS_ACTIVITY: The entity has 2 outgoing 1:M relationship links to DCP_FORMS_ACTIVITY_PRSNL and DCP_FORMS_ACTIVITY_COMP which each show the correctly declared FK, *dcp_forms_activity_id*. However there are 4 incoming 1:M relationships from each of PRSNL, TASK, ENCTR and DCP_FORMS_REF. Appropriate FKs are shown for the first three of these links, but the final link shows the FK *dcp_forms_ref_id* which would appear to be the correct PK attribute name of DCP_FORMS_REF had it been allocated a correctly named PK, however it is instead given the PK name *dcp_form_instance_id,* which is what should appear in DCP_FORMS_ACTIVITY FK list. (PW1)

DCP_FORMS_DEF: DCP_FORMS_REF has the PK *dcp_form_instance_id,* but another attribute *dcp_forms_ref_id* which one would normally expect to be its PK. At the same time entities DCP_FORMS_DEF has the attribute *dcp_forms_ref_id* without it being declared as a FK. No relationship link is shown between the two entities. It has a putatative FK *dcp_section_ref_id* that implies a link to DCP_FORMS_REF. (PW1, PW4)

DCP_FORMS_REF: DCP_FORMS_REF has the PK, with the ill-formed name , *dcp_form_instance_id*. It has a 1:M relation with DCP_FORMS_ACTIVITY. However, the latter does not contain its PK, the attribute *dcp_forms_instance_id* as a FK, instead it contains *dcp_forms_ref_id* as a FK which is declared only as an attribute of DCP_FORMS_REF rather than its PK. It follows then, the PK of DCP_FORMS_REF should be *dcp_forms_ref_id* and not *dcp_form_instance_id*. If this is true, then there is also a missing relationship link with DCP_FORMS_DEF as it too has the attribute *dcp_forms_ref_id*. (PW1, PW2)

It is proposed that the PK of DCP_FORMS_REF should read *dcp_forms_ref_id*. The current description if implemented will not function as would be expected. The effects are that insertions of records into DCP_FORMS_REF will not be linked correctly to DCP_FORMS_ACTIVITY and subsequently to DCP_FORMS_ACTIVITY_COMP and DCP_FORMS_ACTIVITY_PRSNL.

Also any insertions or deletions into DCP_FORMS_ACTIVITY will not cascade appropriately to DCP_FORMS_REF and there will be no data integrity or consistency for data distributed across the two entities. (PW1, PW4).

DCP_INPUT_REF: In DCP_INPUT_REF there appears to be a missing relationship link with DCP_SECTION_REF as both have the attribute *dcp_section_ref_id* although it is not declared as a FK in either entity. Neither entity has a relationship link with any other entity. (PW2, PW4)

DCP_NOMENCATEGORYDEF: There is no PK assigned. There are three possible attributes for the PK, *categorydef_id*, *category_id*, *nomenclature_id* although it is likely to need some concatenated key. None of these attributes are declared as FKs. (PW1, PW2)

DCP_SECTION_REF: DCP_SECTION_REF has the ill-formed name for the PK, *dcp_section_instance_id* along with an attribute *dcp_section_ref_id* which we would normally expect to be the PK of this entity. It has no relationship links to suggest where this PK name might have been inherited from. (PW1, PW4)

L**ONG_BLOB:** LONG_BLOB shares a PK with LONG_BLOB_REFERENCE but no relationship links are shown. (PW4)

LONG_BLOB_REFERENCE: LONG_BLOB_REFERENCE shares the PK, *long_blob_id*, with LONG_BLOB, but no relationship link shown. (PW4)

PIP_COLUMN: FK *pip_section_id* is not declared. There is a missing relationship link with PIP_SECTION via *pip_section_id*. It shows no relationship links shown with any other entity. (PW2, PW4)

PIP_SECTION: FK *pip_id* not declared. There is a missing relationship link with PIP via *pip_id*. (PW2, PW4)

REF_TEXT_VARIATION: FKs, *parent_entity_id* and *parent_entity_name* are not declared. The attributes, *parent_entity_id* and *parent_entity_name* are present in the following entities: REF_TEXT_VARIATION, REF_TEXT_FACILITY_R, DCP_PL_ARGUMENT, DCP_ACTIVITY_LOG, REF_TEXT_RELTN, PIP_PREFS, DCP_FORMS_ACTIVITY_COMP but are not declared as FKs in any of them. Also, the entities that have them as PKs is not shown. The lable "parent_entity" isa highly generic label and so it is difficult to understand which entity in the system it may be referring to. It may in fact be referring to its own entity REF_TEXT_VARIATION. This isa weakness as it is difficult for subsequent users of these diagrams to know what is being referred to. (PW2)

REF_TEXT: There is no PK assigned to REF_TEXT. Perhaps, *refr_text_id* is intended to be the PK (although it is orthographically inconsistent with the second "r"). It shows no relationship links shown with any other entity. However there appears to be a missing relationship link to REF_TEXT_RELTN since it also contains the attribute *refr_text_id*. (PW1, PW2, PW4)

REF_TEXT_RELTN: There is no PK assigned to REF_TEXT_RELTN, but a good candidate could be *ref_text_reltn_id* is declared in the attribute list. FKs are not declared. There is a missing relationship link with REF_TEXT if its PK is indeed *refr_text_id*. (PW1, PW2, PW4)

RISK_ADJUSTMENT_OUTCOMES: The name of the entity is in the plural form which is contrary to the semantics of E-R diagrams.

Patient Management ER Diagram

Dated 8th Nov 2006: Total 84 entities

There are 28 entities with no links to any other entities including these entities which have no PKs, ENCNTR_TYPE_PARAMS, PM_QUE_COLUMN, PM_SCH_LIMIT, PM_QUE_VALUE, PM_SCH_RESULT, PM_SCH_FILTER.

ENCOUNTER: The entity ENCOUNTER has a single attribute *enctr_id* which has 12 1:M relationships and the attribute appears the same in all FKs except in the entity ENCTR_ENCTR_REL which has two relationships with ENCOUNTER represented by the attributes *encntr_id* and *related_encntr_id* with both as FKs.

ENCNTR_PENDING: PRSNL is shown with a 1:M relationship to ENCNTR_PENDING but it has no attribute *person_id*. There is an attribute *pending_prsnl_id* that might be an equivalent attribute but it is not listed as a FK. (PW2)

ENCTR_PENDING_HIST: PRSNL is shown with a 1:M relationship to ENCTR_PENDING_HIST but it has no attribute *person_id*. There is an attribute *pending_prsnl_id* that might be an equivalent but it is not listed as a FK. (PW2)

ENCNTR_TYPE_PARAMS: No PK nor FKs declared. There is a missing relationship link with ORGANIZATION via organization_id. (PW1, PW4)

FLEX_QUERY: FK, *long_text_id*, not declared. There is a missing relationship link with LONG_TEXT via *long_text_id*. There are no relationship links shown with any entity. (PW2, PW4)

GROUP_ENTITY_RELTN: There are no relationship links shown with any entity and FKs for *group_id* and *related_id* not declared. (PW2)

HP_ALIAS_POOL_RELTN Entity: The entity CODE_VALUE has a single attribute *code_value* as the PK. It has a 1:M relationship with the entity HP_ALIAS_POOL_RELTN. There is no equivalently named attribute in HP_ALIAS_POOL_RELTN. It has a FK *location_cd* but no link, which might be meant to be linked to LOCATION entity on the Patient Tracking ERD. (PW2)

LONG_TEXT_REFERENCE: LONG_TEXT_REFERENCE uses the PK, *long_text_id,* from LONG_TEXT without declaring it as a FK. (PW1)

LONG_BLOB_REFERENCE: LONG_BLOB_REFERENCE has a PK, *long_blob_id*, which is the PK of LONG_BLOB, an entity that is present in the "Document Management" ERD. It is not declared as an FK in this ERD. (PW1)

ORG_SET_PRSNL_R: This entity has M:1 relationship link to PRSNL, however, the only FK is *prsnl_id* in this entity and hence this could indicate the PK of PRSNL should not be *person_id* but *prsnl_id.* Neither the FK is declared nor the relationship link shown with *the* entity ORG_SET for the attribute *org_set_id*. (PW1, PW2, PW4)

PM_DOC_DISTRIBUTION: PM_DOC_DISTRIBUTION **s**hares the PK, *distribution_id*, with PM_QUE_DISTRIBUTION which is not declared as FK. Datatype for the same attribute, *distribution_desc* is different in both entities with it being CHAR(100) in this one and VARCHAR2(100) in PM_QUE_DISTRIBUTION. ( )

PM_FLX_ACTION: This entity has no relationship links and no FKs are declared. There appear to be missing relationship links with PM_FLX_DATA_SOURCE and LONG_TEXT via attributes *data_source_id* and *info_long_text_id* respectively. (PW2, PW4)

PM_FLX_CONVERSATION: It appears FKs have not been declared for attributes *icon_long_text_id*, *info_log_text_id*, and *rule_long_text_id*. These attributes could indicate missing relationship links to LONG_TEXT/LONG_TEXT_REFERENCE as both have the same PK. However, it can be seen that there exists a ternary relationship entity involving PM_FLX_CONVERSATION , LONG_TEXT-REFERENCE and LONG_BLOB_REFERENCE through the entity PM_FLX_CONV_OPTIONS. If this is true, then why are these FKs present in PM_FLX_CONVERSATION. Or do these keys indicate links to LONG_TEXT and LONG_BLOB_REFERENCE? (PW2, PW3)

PM_FLX_CONV_OPTIONS: This seems to be a ternary relationship involving PM_FLX_CONVERSATION, LONG_TEXT_REFERENCE and LONG_BLOB_REFERENCE. However, it contains no FKs similar to *long_text_id* and *long_blob_id* instead it has undeclared putative FK candidates *value_long_text_ref_id* and *value_blob_ref_id*. It follows then that the PKs of LONG_TEXT_REFERENCE and LONG_BLOB_REFERENCE should perhaps be *long_text_ref_id* and *long_blob_ref_id* respectively. This would also solve the problem of the shared PKs that these entities have. (PW1, PW2)

PM_FLX_DATA_SOURCE: PM_FLX_DATA_SOURCE has no relationship links to other entities and putative FKs, *parent_entity_id* and *info_long_text_id*, are not declared. It appears there is a missing relationship link with LONG_TEXT with attribute *info_long_text_id* and with an unknown entity through attribute *parent_entity_id.* (PW4)

PM_FLX_TASK_CONV_RELTN: PM_FLX_TASK_CONV_RELTN is a disconnected entity, with no declared FKs. There appear to to be missing relationship links with ORGANIZATION and PM_FLX_CONVERSATION via attributes *organization_id* and *conversation_id* respectively. (PW2, PW4)

PM_HIST_TRACKING: There is no FK declaration for *create_prsnl_id* for apparent link to PRSNL. (PW2, PW4)

PM_INFO_HIST: The FK, *parent_entity_id* is not declared, and there are no relationship links to other entities. (PW1, PW4)

PM_LOCK: There are no relationship links to other entities and the FKs, *person_id*, *user_id*, are not declared. There is a missing relationship link with PERSON. (PW2, PW4)

PM_LOC_ATTRIB: There are no relationship links to other entities, and the FK, *parent_entity_id* not declared. (PW2, PW4)

PM_LOC_ATTRIB_HIST: There are no relationship links to other entities, and the FKs, *pm_loc_attrib_id*, *parenty_entity_id*, *value_id* are not declared. There is a missing relationship link to PM_LOC_ATTRIB. (PW2, PW4)

PM_PREF: The putatative FKs, *icon_long_blob_id*, *details_long_text_id*, *active_status_prsnl_id* and *data_status_prsnl_id* are not declared. There appear to be missing relationship links with LONG_BLOB_REFERENCE and LONG_TEXT/ LONG_TEXT_REFERENCE via the attributes *icon_blob_id* and *details_long_text_id* respectively. The attributes, *active_status_prsnl_id* and *data_status_prsnl_id* are present in several entities but are not declared as FKs. (PW2, PW4)

PM_PREF_SETUP: PRSNL is shown with a 1:M relationship to PM_PREF_SETUP but it has no attribute *person_id* although it does have a seemingly equivalent attribute in *person_prsnl_id* as a FK, and hence it follows that the correct PK for PRSNL would be *prsnl_id* and not *person_id*. (PW2)

PM_QST_ANSWER: There are no relationship links to other entities and FKs *question_id*, and *parent_entity_id* are not declared. There is a missing relationship link to PM_QST_QUESTION via *question_id*. (PW2, PW4)

PM_QST_CHOICE: There are no relationship links to other entities and FK not declared for *question_id*. There is a missing relationship link to PM_QST_QUESTION via *question_id*. (PW2, PW4)

PM_QST_QUESTION: There are no relationship links to other entities and no declared FKs. There is a missing relationship link to PM_QST_QUESTIONNAIRE via *questionnaire_id*. (PW2, PW4)

PM_QUE_COLUMN: There is no PK, no relationship links, and a seeming FK not declared for *work_list_id*. There is a missing relationship link to PM_QUE_WORK_LIST via *work_list_id*. (PW2, PW4)

PM_QUE_DISTRIBUTION: PM_QUE_DISTRIBUTION shares the PK, *distribution_id,* with PM_DOC_DISTRIBUTION but no relationship links are shown. (PW4)

PM_QUE_DIST_FILTER: There are no relationship links to other entities and FK, *distribution_id* is not declared. There is a seemingly missing relationship link to PM_DOC_DISTRIBUTION/PM_QUE_DISTRIBUTION as both have the same PK. (PW2, PW4)

PM_QUE_ITEM_DIST: There are no relationship links to other entities and FKs, *work_item_id*, *distribution_id*, are not declared. There are seemingly missing relationship links to PM_QUE_WORK_ITEM and PM_DOC_DISTRIBUTION/ PM_QUE_DISTRIBUTION via *work_item_id* and *distribution_id* respectively. (PW2, PW4)

PM_QUE_METHOD: There are no relationship links to other entities, and the FK, *application_id*, is not declared. There is a seemingly missing relationship link to PM_QUE_APPLICATION via *application_id*. (PW2, PW4)

PM_QUE_PROPERTY: There are no relationship links to other entities and the FK, *parent_id*, is not declared. (PW2)

PM_QUE_VALUE: There are no relationship links to other entities, no PK, and apparent FKs are not declared. There are apparent missing relationship links to PM_QUE_WORK_LIST and PM_QUE_PROPERTY via *work_list_id* and *property_id* respectively. (PW2, PW4).

PM_QUE_WORK_LIST: There are no relationship links to other entities and the FK, *method_id* is not declared. There is a missing relationship link to PM_QUE_METHOD. (PW4)

PM_RPT_DISPLAY: There are no PKs. No relationship links to other entities and the FK, *parent_display_id* is declared. (PW1, PW2)

PM_SCH_LIMIT: There are no PKs, no relationship links to other entities, and the FK is not declared for attribute *setup_id*. There is a missing relationship link to PM_SCH_SETUP. (PW1, PW2, PW4)

PM_SCH_PREFERENCES: There are no relationship links to other entities and FKs for *person_id* and *application_number* not declared. In the schedule, "security application", it was noted that the entity APPLICATION should have the attribute *application_number* as its PK. Hence, *application_number* in this entity is perhaps a FK . There is a missing relationship link with PERSON via *person_id*. (PW2, PW4)

PM_TRANSFER_ENCNTR_RELTN: PRSNL is shown with two M:1 relationships to PM_TRANSFER_ENCNTR_RELTN but it has no attribute *person_id.* However it could be linked twice to PRSNL via *prsnl_person_id* and *transfer_prsnl_person_id*. From previous logic, the correct PK for PRSNL seems to be *prsnl_id* and not *person_id*. Alternatively, there is the question: Is this entity linked correctly to PRSNL or should it be linked to PERSON instead?

PM_USER_DEFINED_HIST: FKs , *parent_entity_id*, *value_long_text_id* are not declared. There is an apparent missing relationship link with LONG_TEXT_REFERENCE through attribute *value_long_text_ref_id* . (PW 2, PW4)

PM_WAIT_LIST_FUTURE_STATUS: The apparent FK not declared and perhaps should be linked with LONG_TEXT via *comment_long_text_id*. (PW2)

PM_WAIT_LIST_HIST: The apparent FKs *admitting_prsnl_id*, *active_status_prsnal_id*, and *chg_id* are not declared and perhaps should have relationship links PRSNL. It is unclear the entity from which *chg_id* derives. (PW2)

PM_WORK_QUEUE: This entity creates a relationship between the PERSON and ENCOUNTER entities having a M:1 relationship with each of them with the appropriate declared FKs, *person_id* an *enctr_id*. There is a missing relationship link with PM_QUE_WORK_ITEM via the putative undeclared FK *work_item_id*. The attribute, *completed_by_id* would seemingly require the PERSON/PRSNL identifier. (PW1, PW2)

PRSNL: The correct PK should apparently be *prsnl_id* and not *person_id*. The PK *person_id* is shared with PERSON and hence is likely to cause significant confusion. A possible explanation for such modeling would be to identify staff (PRSNL) when they are patients (PERSONS). (PW1)

Separating PERSON and PRSNL references

There are a total of 66 attributes with the string "_prsnl_id" and all are part of the attribute name *active_status_prsnl_id* except *pending_prsnl_id* (2), *person_prsnl_id*, *admitting_prsnl_id*, *create_prsnl_id*, *first_prsnl_id*, and *second_prsnl_id*. In only two instances are these listed as FKs in entity ORG_SET_PRSNL_R, and entity PM_PREF_SETUP.

There are a total of 14 attributes with the string "person_id". In two entities it is the PK, namely PRSNL and PERSON. It is present in PM_LOCK and PM_SCH_PREFERENCES as an attribute but not as a PK or FK - both entities are shown with no links diagrammatically and no FKs and PM_SCH_PREFERENCES is shown without a PK.

In FAMILY_RELATION_EXCEPTION “person_id” is used in the attribute *suggested_related_person_id* and declared as a FK.

In PM_TRANSFER_ENCNTR_RELTN it is used in the attribute *prsnl_person_id* and listed as a FK. Semantically this attribute name is a concern as its meaning at the surface level suggests it is the “identifier” of a “person” who is a member of the “personnel”, but this would normally be expected to be identified by the attribute *prsnl_id*. The potential for confusion is substantial and dangerous given the already documented confusion over the appropriate PK for the entity PRSNL apropos the entity PERSON.

In PM_TRANSFER_ENCNTR_RELTN it is used in the attribute *transfer_prsnl_person_id* and listed as a FK.

It is present as a FK in the remaining 7 entities.

PATIENT TRACKING ER Diagram

Dated 8th Nov 2006. Total 60 entities.

This ERD poses a general question about the naming of entities. A collection of entities have the word TRACK and another group use the word TRACKING. The level of similarity of the meaning of the two words suggests that one is the abbreviation of the other. This would be a poor design as it makes it difficult to understand the semantic differences and similarities that are meant to be captured in the design.

FN_OMF_ENCNTR: The entity TRACK_REFERENCE with a PK of *tracking_ref_id* has THREE 1:M relationships with the entity FN_OMF_ENCNTR. There is no attribute in the FN_OMF_ENCNTR entity that is a match for the attribute *tracking_ref_id*, but there is an attribute *tracking_id* as a FK which may be an intended match. The two entities have a common FK of *tracking_group_cd* which is the only self-evident basis on which they can be connected. The attribute *tracking_group_cd* is present in several entities wherein it is declared as an FK but in FN_OMF_ENCNTR it is not marked as a FK. Similarly, the attributes *loc_nurse_unit_cd* and *loc_room_cd* are marked as FKs in TRACKING_LOCATOR but not in FN_OMF_ENCNTR. (PW2)

MRU_LOOKUP_FOLLOWUP: The apparent FK *parent_entity_id* is not declared as such and no relationships are shown for it. (PW2, PW4)

MRU_PRIMARY_FACT_EXTENSION: This entity has no relationship links shown with any other entities and the apparent FK *primary_fact_id* is not declared as such. (PW2, PW4)

MRU_STATISTIC: This entity has no relationship link with any other entity and the apparent FKs *primary_fact_id*, *prsnl_id*, and *prsnl_community_id* are not declared as such. (PW2, PW4)

PAT_ED_DOCUMENT: PAT_ED_DOCUMENT has 1:M relationships with both PAT_ED_DOC_ACTIVITY and PAT_ED_DOC_LEAFLET_ACTIVITY but its PK cannot be found in their attribute list. However, there is an attribute *pat_ed_doc_id* as a FK which may be an intended match. (PW1, PW2, PW3)

PAT_ED_DOC_ACTIVITY: This is a relationship between PAT_ED_RELTN and PAT_ED_DOCUMENT entities with a M:1 link to each. The FK to PAT_ED_DOCUMENT is declared to be the attribute *pat_ed_doc_id* whereas the PK of PAT_ED_DOCUMENT is *pat_ed_document_id*. The attribute *refr_text_id* is an undeclared FK without any links shown. (PW2, PW3)

PAT_ED_DOC_FOLLOWUP: The FKs *provider_id* and *location_cd* are not declared as such. There is a missing relationship link to LOCATION via *location_cd*. (PW2, PW4)

PAT_ED_DOC_LEAFLET_ACTIVITY: This entity has a declared FK *pat_ed_doc_id* which should most likely be *pat_ed_document_id* to represent the M:1 relationship link to PAT_ED_DOCUMENT.

TRACK_COMPLAINT: The apparent FKs *parent_complaint_id* and *foreign_complaint_id* are not declared. This entity has no relationship links shown with any other entity. (FW2, FW4)

TRACK_EVENT: TRACK_EVENT entity with PK *track_event_id* has a 1:M relationship with itself. There is no attribute with a similar name except *def_next_event_id*.(PW3)

TRACK_GROUP : TRACK_GROUP has no declared PK but has an attribute *tracking_group_cd* which might be the intended PK. (PW1)

TRACK_LOCATION_PARAM: There is a missing relationship link to LOCATION for the PK *location_id*. There are no relationship links shown with any entity. (PW2, PW4)

TRACK_ORD_EVENT_RELTN: There is no declared PK. The attribute, *track_event_id*, is an apparent FK entity, with also a missing relationship link to TRACK_EVENT via *track_event_id*. (PW1, PW2, PW4)

TRACK_PREFS: This entity has no relationship link with any other entity and the apparent FK *parent_pref_id* is not declared as such. (PW2, PW4)

TRACK_REFERENCE: The name of the PK has been poorly defined as *tracking_ref_id* whereas it should be *track_reference_id*. This entity has single 1:M relationships with TRACKING_PRSNL (1) and TRACKING_LOCATOR (1) but neither of these entities have *tracking_ref_id* as a FK in them. Likewise, none of the multiple links with TRACKING_CHECKIN (5) and FN_OMF_ENCNTR (3) have declared FKs. (PW1, PW2, PW3)

TRACK_SENSOR: This entity has no relationship links shown with any other entity. The apparent FK, *location_cd* is not declared. The relationship link to LOCATION via location_cd is missing. (PW2, PW4)

TRACK_TRIGGERS: This entity is named in the plural which is against normal practice. It contains an apparent FK, *conversation_id* which is not linked to a likely entity. (PW2)

TRACKING_CHECKIN: The apparent FKs *parent_entity_id* and *reactive_user_id* are not declared. The entity TRACK_REFERENCE with a PK of *tracking_ref_id* has FIVE 1:M relationships with this entity. There is no attribute in TRACKING_CHECKIN that is a match for the attribute *tracking_ref_id*. The two entities have a common FK of *tracking_group_cd* which is the only self-evident basis on which they can be connected. TRACKING_CHECKIN has 16 FK attributes with only 6 incoming 1:M links. (PW2, PW4)

TRACKING_ENCNTR_EXTENSION: TRACKING_ENCNTR_EXTENSION which has 3 attributes labelled as FKs, *encntr_id*, *pat_ed_acknowledged_user_id*, and *dc_signed_user_id* but no incoming links to indicate the source entities for these FKs. (PW2)

TRACKING_ENCNTR_PRSNL_RELTN: The entity TRACKING_PRSNL has the PK *tracking_enctr_prsnl_reltn_id*. The attribute *tracking_id* is the PK of the entity TRACKING_ITEM and is not declared as a FK. TRACKING_ITEM has two 1:M relationships with the entity TRACKING_ENCNTR_PRSNL_RELTN. Given that the attribute *tracking_id* is present in the entity but not a FK, then two other attributes may be the intended links as they are listed as FKs *encntr_tracking_id*, *prsnl_tracking_id*. (PW2)

TRACKING_EVENT: The apparent FKs *requested_id, onset_id, complete_id, inlab_id, collected_id,* and *comonrew_id* are not declared. Also, there may be a spelling conflict where the attribute *complete_ind* of TRACK_REFERENCE is intended to be the same as the attribute *complete_id* in TRACKING_EVENT and just spelt inconsistently. (PW2)

TRACKING_ITEM: TRACKING_ITEM with a PK of *tracking_id* has thirteen 1:M relationships, and 6 FKs of which only two show links on the diagram. They are LOCATION with a PK of *location_cd* which probably matches the FK *loc_cd*, and the entity ITEM_MASTER with the PK *item_id* which has no apparent match to a FK or any other attribute. The apparent FKs, *start_tracking_id*, *end_tracking_id*, and *parent_entity_id* are not declared as such. The attribute *active_status_cd* is marked as a FK here but inconsistently it is not marked as a FK in TRACK_ACTION_LINK, TRACK_COLLECTION, TRACK_COLLECTION_ELEMENT and TRACKING_COMMENT. The presence of the attributes *start_tracking_id* and *end_tracking_id* may indicate relationship links with itself. It is linked twice to TRACKING_ENCNTR_PRSNL_RELTN and the links seem to be via *encntr_tracking_id* and *prsnl_tracking_id* which are marked as FKs. However, there is a presence of the attribute *tracking_id* as well in the above entity which is the PK of TRACKIN_ITEM. This means, there should be one more link with the above entity. (PW2, PW4)

TRACKING_LOC_STATUS: This entity has no relationship links shown with any other entity. The apparent FKs, *location_cd, doc_status_evt_id, nurse_status_evt_id, lab_status_evt_id, loc_status_evt_id,* and *loc_status_item_id* are not declared. There is a missing relationship link to LOCATION via *location_cd*. (PW2, PW3)

TRACKING_LOCATOR: The entity TRACK_REFERENCE with a PK of *tracking_ref_id* has a 1:M relationship with the entity TRACKING_LOCATOR. There is no attribute in the TRACKING_LOCATOR entity that appears to be a match for the attribute *tracking_ref_id*. The two entities have no common FK so there is no obvious manner in which they can be connected. The

apparent FK, *unavail_tracking_event_id* is not declared. The data type for *locator_create_date* is given as FLOAT but should be DATE. (PW2)

TRACKING_PREARRIVAL: The TRACKING_PREARRIVAL entity has 4 FKs *attached_encntr_id*, *attached_person_id*, *pcp_provider_id, reg_prsnl_id* but shows no incoming links that indicate the source entities for the references. (PW4)

TRACKING_PRSNL: An entity TRACK_REFERENCE with a PK of *tracking_ref_id* has a 1:M relationship with TRACKING_PRSNL. There is no attribute in TRACKING_PRSNL entity that appears to be a match for the attribute *tracking_ref_id*. The two entities have a common FK of *tracking_group_cd* which is the only self-evident basis on which they can be connected.(PW3)

TRACKING_PRSNL_REF: this entity has 3 FKs but only two have links to other entities. The FK attribute *tracking_ref_type_cd* appears to not have a matching relationship link. (PW4)

TRACKING_PRV_RELN: The entity has no relationship links shown with any other entity. The FKs *tracking_id*, and *tracking_provider_id* are not declared. There is a missing relationship link to TRACKING_ITEM for the attribute *tracking_id*. (PW2, PW4)

DCP Power Chart ER Diagram

Dated 8th Nov 2006. Total 35 entities.

APP_PREFS: This entity has no relationship links to any other entity but does have an attribute *prsnl_id* proposing a link to the PRSNL entity but it is not declared as a FK. (pw2, pw4)

CHART_DEFINITION: The entity REF_DATASTATS_HIST contains the attribute *chart_definition_id* which is the PK of CHART_DEFINITION, although it is not labelled as a FK in the former. It follows then that the 2 entities must be connected but the link is missing. (PW2, PW4)

CHART_DEFINITION_HIST: This entity shares the PK *action_seq* with REF_DATASTATS_HIST and REF_DATAPOINT_HIST, but there are no relationship links and no FKs declaration. (PW2, PW4)

CONFIG_PREFS: There is no PK and no relationship with any other entities. The attribute *config_prefs_id* should perhaps be the PK for this entity. (PW1)

DCP_CUSTOM_COLUMNS : The Entity PRSNL with the PK attribute *person_id* has a 1:M relationship with the entity DCP_CUSTOM_COLUMNS which has only one FK shown as *prsnl_id*. (PW2)

DETAIL_PREFS: This entity has no relationship links to any other entity but does have an attribute *prsnl_id* suggesting a link to the PRSNL entity but it is not declared as a FK. It also has the attribute *person_id* suggesting a link to either the PERSON or PRSNL entities but it is not declared as a FK. The attribute *application_number* should probably be an FK coming from the entity APPLICATION in the Security ERD.

NAME_VALUE_PREFS: The apparent FK, *parent_entity_id*, is not declared, and there are no relationships with any other entities. (PW1)

ORDER_CATALOG_SYNONYM: The apparent FKs, *oe_format_id*, *order_sentence_id*, and *item_id* are not declared. Concomitantly there are then missing relationship links to the respective entities. (PW2, PW4)

PERSON_PRSNL_ACTIVITY: This entity has both attributes, *prsnl_id* and *person_id* but the former is not declared as FK while the latter is. There is a relationship link to PERSON but not to PRSNL. This is a further indication that the PK for PRSNL should be *prsnl_id* and not *person_id*. The semantics of this entity is also hard to reconcile as you would conjecture

that it is a about a personnel activity on a person (patient?) but it is not liked to any other entity in the system. (PW1, PW2, PW4)

PROXY: There is a M:1 relationship link to entity PRSNL_GROUP_RELTN but its PK, *prsnl_group_reltn_id* is not present in PROXY. There is a missing relationship link to PERSON/PRSNL since the attribute *person_id* is present in the entity but undeclared as a FK. Likewise for the attribute *proxy_person_id*.(PW2, PW4)

PRSNL: PRSNL has a 1:M relation with DCP_CUSTOM_COLUMNS. However, the latter does not have the attribute *person_id*, instead it has *prsnl_id*, suggesting that *prsnl_id* is the correct PK for PRSNL. Also, PRSNL shares the PK, ostensibly *person_id* with entity PERSON. Semantically, *person_id* seems correct for PERSON; however, PRSNL should have PK as *prsnl_id*. (PW1)

PRSNL_GROUP_RELTN: Has a 1:M relation with PROXY; however, the latter does not have the PK of PRSNL_GROUP_RELTN, *prsnl_group_reltn_id*, as FK. (nil)

REF_DATAPOINT_HIST: This entity shares the PK *action_seq* with CHART_DEFINTION_HIST and REF_DATASTATS_HIST. The FK, *ref_dataset_id*, is not declared, although the relationship link is transitive through REF_DATA_POINT . (PW2, PW4)

REF_DATASTATS_HIST: This entity shares the PK *action_seq* with CHART_DEFINITION and REF_DATAPOINT_HIST. The FK, *chart_defintion_id* is not declared. (PW2, PW4)

VIEW_COMP_PREFS: This entity has no relationship links to any other entity but does have an attribute *prsnl_id* suggesting a link to the PRSNL entity but it is not declared as a FK.

VIEW_PREFS: This entity has no relationship links to any other entity but does have an attribute *prsnl_id* suggesting a link to the PRSNL entity but it is not declared as a FK.

Security Application ER Diagram

Dated 27th May 2009. Total 33 entities

ACCESS_CONTROL_POLICY: The following attributes have duplicate entries in the ERD *updt_dt_tm, ipdt_id, updt_task, updt_cnt, updt_applctx.*

APPBAR_SECURITY: This entity has 2 PK attributes that are of the VARCHAR data type, *name_group*, and *member*. It has a name as a partial PK, *name_group*, which is normally undesirable as it may cause ambiguity in addition to loss of preservation of uniqueness. There are no relationship links shown with any other entities. (PW1, PW4)

APPLICATION: The PK is labelled as *def_sched_id* which violates naming conventions as one would expect *application_id*. Moreover, it has a 1:M relation with the entity APPLICATION_TASK_R which has the attribute *application_number* as a FK but no *def_sched_id* as an attribute. It appears likely that the entry in the entity APPLICATION for the PK is incorrect and should be *application_number,* although this is still ill-formed. (PW1)

APPLICATION_ACCESS: The apparent FK, *application_number* is not declared. There are no relationship links shown with any other entities. The attribute *application_number*, supposedly should be the PK for the entity APPLICATION and hence these 2 entities must be connected although there are no relationship links shown. (PW2, PW4)

APPLICATION_CONTEXT: This entity has a potentially ill-formed PK name is *hna_reconcile_except_id* which normally would be *application_concept_id.* Apparent FKs, *except_group_id*, and *person_id,* are not declared. There are no relationship links shown with any other entities. (PW1, PW2, PW4)

APPLICATION_INI: The FKs *application_number*, and *person_id* are not declared. There are no relationship links shown with any other entities. We know that *application_number* is a FK and probably is the PK of the entity APPLICATION and if that is correct, then the 2 entities should be connected, which is not the case presently. (PW1, PW2, PW3)

APPLICATION_PARAMETER: This entity has a composite PK, *clinical_service_reltn_id* and *parent_entity_Name*, the latter not being an integer can readily cause ambiguity and loss of uniqueness. Moreover, the PK component *parent_entity_name* is not semantically consistent with the name of the entity APPLICATION_PARAMETER. There are no relationship links shown with any other entities. (PW1, PW2, PW4)

APPLICATION_TASK: APPLICATION_TASK with PK, *task_number*, has 1:M relationships with TASK_ACCESS, APPLICATION_TASK_R and TASK_REQUEST_R giving it the appearance of being a ternary relationship link . However, excepting APPLICATION_TASK_R, the other 2 entities do not have the attribute *task_number.* However, there is an attribute *nomenclature_id* as a FK in TASK_ACCESS. If it has such a role then it is poorly named and should have the name APPLICATION_TASK_RELN. It appears likely that the entry in the entity APPLICATION_TASK for the PK is not being modeled correctly but it is difficult to suggest the appropriate remedy. (PW1)

APPLICATION_TASK_R: The partial PK, *application_number* is derived from APPLICATION assuming that the PK in APPLICATION is incorrectly labelled. This entity does not seem to be serving the role of a relationship between two entities and so is mislabeled with the "_R" at the end of its name.(PW1)

APPLICATION_INI: There is no declared PK. It has the putative FK attribute *application_number* but it is undeclared. There are no relationship links shown to any other entities. (PW1)

CPMPROCESS_ERROR: The apparent FKs, *que_id*, *destination_step_id*, are not declared. There are no relationship links shown with any other entities. This entity shares the PK *request_number* with 2 other entities: PUBLISHABLE_REQUEST and REQUEST. It should be linked to CPMPROCESS_QUE via the attribute *que_id*. (PW1, PW2, PW4)

CPMPROCESS_QUE: The apparent FKs, *request_number*, and *next_que_id*, are not declared. There are no relationship links shown with any other entities. It has an attribute *request_number* which is a PK for supposedly 3 entities: PUBLISHABLE_REQUEST, CPMPROCESS_ERROR and REQUEST. Hence, CPMPROCESS_QUE must be linked to one of the above entities that has the original PK. Moreover, it should also be linked with CPMPROCESS_ERROR, which has an attribute *que_id* although not declared as FK, and which is the PK for CPMPROCESS_QUE. (PW1, PW2, PW4)

ENCOUNTER: The ENCOUNTER entity has the PK of *pathway_catalog_id* which is an malformed named for this entity, but in another system ERD "Task List" it has its PK as the well-formed name *encntr_id* . Moreover, this entity has a 1:M relation with the entity ENCNTR_LR_RELTN that has *encntr_id* as a FK and no attribute as *pathway_catalog_id*. This reinforces the evidence that this PK is an error. (PW1)

ENTITY_LOCK: There is no declared PK. The FKs, *entity_id, lock_prsnl_id, unlock_prsnl_id,* and *lockkey_id*, are not declared. There are no relationship links shown with any other entities. (PW1, PW2, PW4)

ORGANIZATION: This entity has the PK *encntr_id* which is a malformed name, which should be *organization_id*. However *encntr_id* is the well-formed PK name for the entity ENCOUNTER in another system ERD "Task List". ORGANIZATION is connected to WORKGROUP_LR_MAPPING with a 1:M relationship; however, the latter has *org_id* as a FK and not *encntr_id*. It follows that the PK for ORGANIZATION should have been *org_id*. This would however be a slightly malformed name, the correct form being *organisation_id*. (PW1)

PERSON: The PK is shown as *reference_task_id* which is malformed and should be *person_id*. This is evident from the fact that PERSON has a 1:M relation with APPLICATION_GROUP and the latter does not have *reference_task_id* as a FK but instead has *person_id* as a FK. PERSON is also shown with the PK *person_id* on other ERDs. It follows that the PK name is an error. (PW1)

PUBLISHABLE_REQUEST: This entity shares the PK, *request_number,* with 2 other entities: CPMPROCESS_ERROR and REQUEST. There are no relationship links shown with any other entities. (PW4)

REQUEST: The entity REQUEST with an attribute *request_number* as a PK has a 1:M relationship with an entity REQUEST_PROCESSING. There is no attribute in REQUEST_PROCESSING entity that appears to be a match for the *request_number* attribute but there is an attribute *policy_text_id* as a FK. It is difficult to detect a semantic commonality between the two names that would make one believe they are intended to be the same attribute, so there is no ready explanation for the disparity here. It follows that if REQUEST_PROCESSING did have the attribute *request_number*, which of the 3 entities that share the same PK, PUBLISHABLE_REQUEST, CPMPROCESS_ERROR and REQUEST, should it actually be identified with? Only one of the 3 can have *request_number* solely as PK, while the remainder should have it as FK. (PW1, PW2, PW4)

SEC_PRINCIPAL: This entity contains an attribute with an apparent unidentified FK, *parent_entity_id*. (PW2)

TASK_ACCESS: This entity is in a M:1 relationship with APPLICATION_TASK and so the PK of *task_number* would be expected to be present as a FK. Hence, the PK component *nomen_entity_reltn_id* appears to be incorrect or should be a FK. There is a missing relationship link with the entity that has *nomenclature_id* as a PK. The apparent FKs *parent_entity_id* and *child_entity_id* are not declared. (PW1, PW2, PW3)

TASK_REQUEST_R: Both the PK attributes, *org_set_id*, *organisation_id*, are shown as FKs but no relationship links are shown to obvious matching entities. Considering that the PK for the entity ORGANIZATION should have been *organization_id*, there is a missing relationship link between ORGANIZATION and TASK_REQUEST_R. There is a link to entity APPLICATION_TASK but TASK_REQUEST_R does not carry its PK, *task_number* nor does there seem to be a synonymic attribute name. (PW1, PW2, PW4)

Review of Primary Key Names

There are 18 entities with PK names inconsistent with the entity name. There are another 12 entities on the ERDs that have consistent naming and 2 that have no PK name (see Table 2).

Entity	PK Name	Entity	PK Name
ACCESS_CONTROL_POLICY	*encntr_prsnl_reltn_id*	**ENCOUNTER**	*pathway_catalogue_id*
APPBAR_SECURITY	*empi_name_thesaurus_id , name_group,member*	ENTITY_LOCK	*lock_seq_id**
APPLICATION	*def_sched_id*	**ORGANIZATION**	*encntr_id*
APPLICATION_CONTEXT	*hna_reconcile_except_id*	**PERSON**	*reference_task_id*
APPLICATION_PARAMETER	*clinical_service_reltn_id*	PUBLISHABLE_REQUEST	*request_number*
APPLICATION_TASK	*task_number*	SEC_POLICY	*flex_data_source_id*
APPLICATION_TASK_R	*application_number, task_number.* This would be correct if APPLICATION was consistently named.	TASK_ACCESS	*nomen_entity_reltn_id, nomenclature_id*
COMPROCESS_QUE	*que_id**	TASK_REQUST_R	*org_st_id , organization_id*
CPMPROCESS_ERROR	*request_number, error_id*	REQUEST_PROCESSING	*consent_policy_id, policy_id*

Table 2. List of entities from the Security Application ERD which have PK names that have some inconsistency with the entity name. * indicates there is some implied association between the entity name and PK name but it is not entirely convincing. Major entities used across many ERDs are shown in bold.

Appendix 2 - Details of Analysis of ERDs

Figure 1. Redrawn ERD showing weaknesses in the Document Management Design

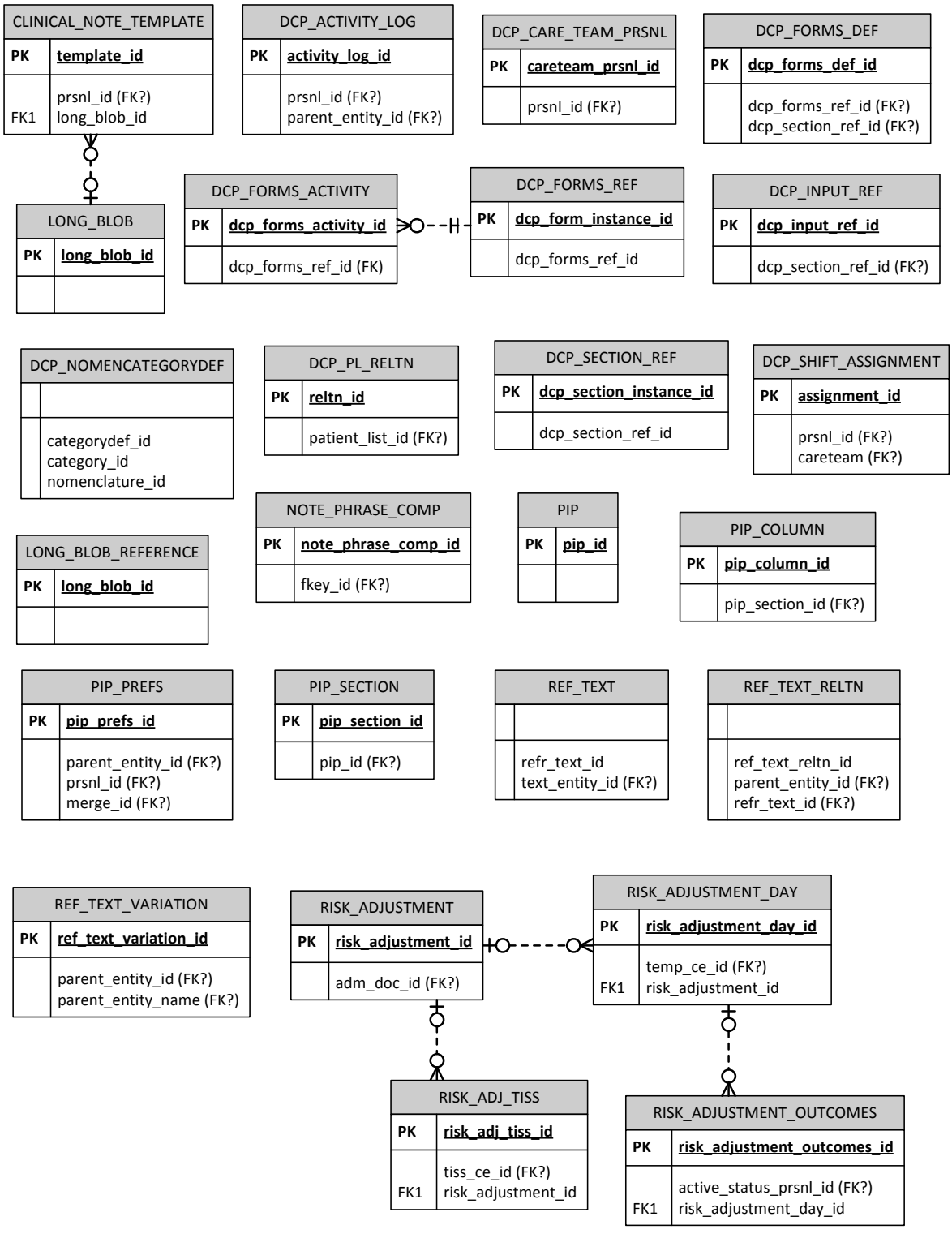
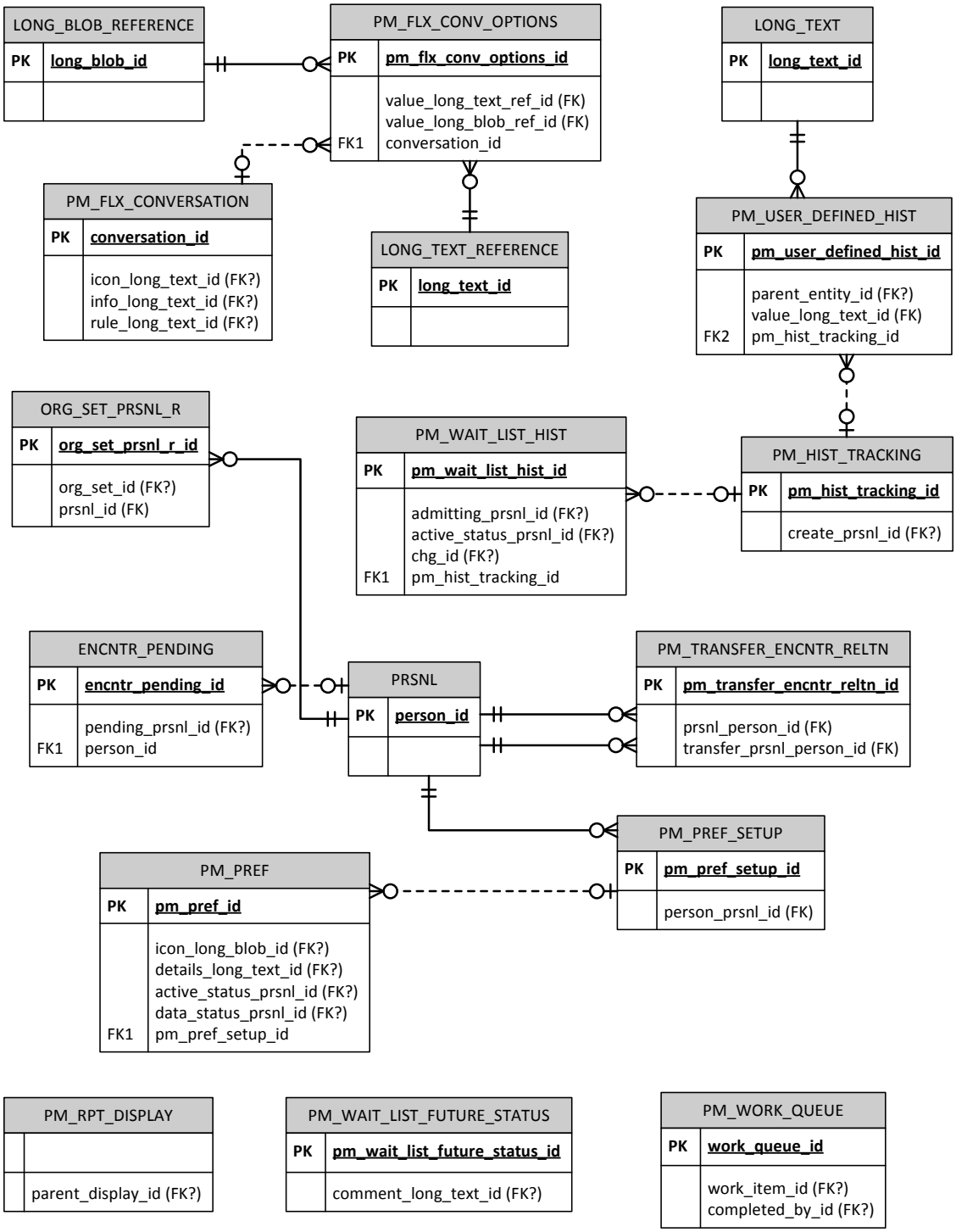


Figure 2. Redrawn ERD showing weaknesses in the Patient Management ERD Design - Pt 1



Patient Management EDR Design Weaknesses - Pt2

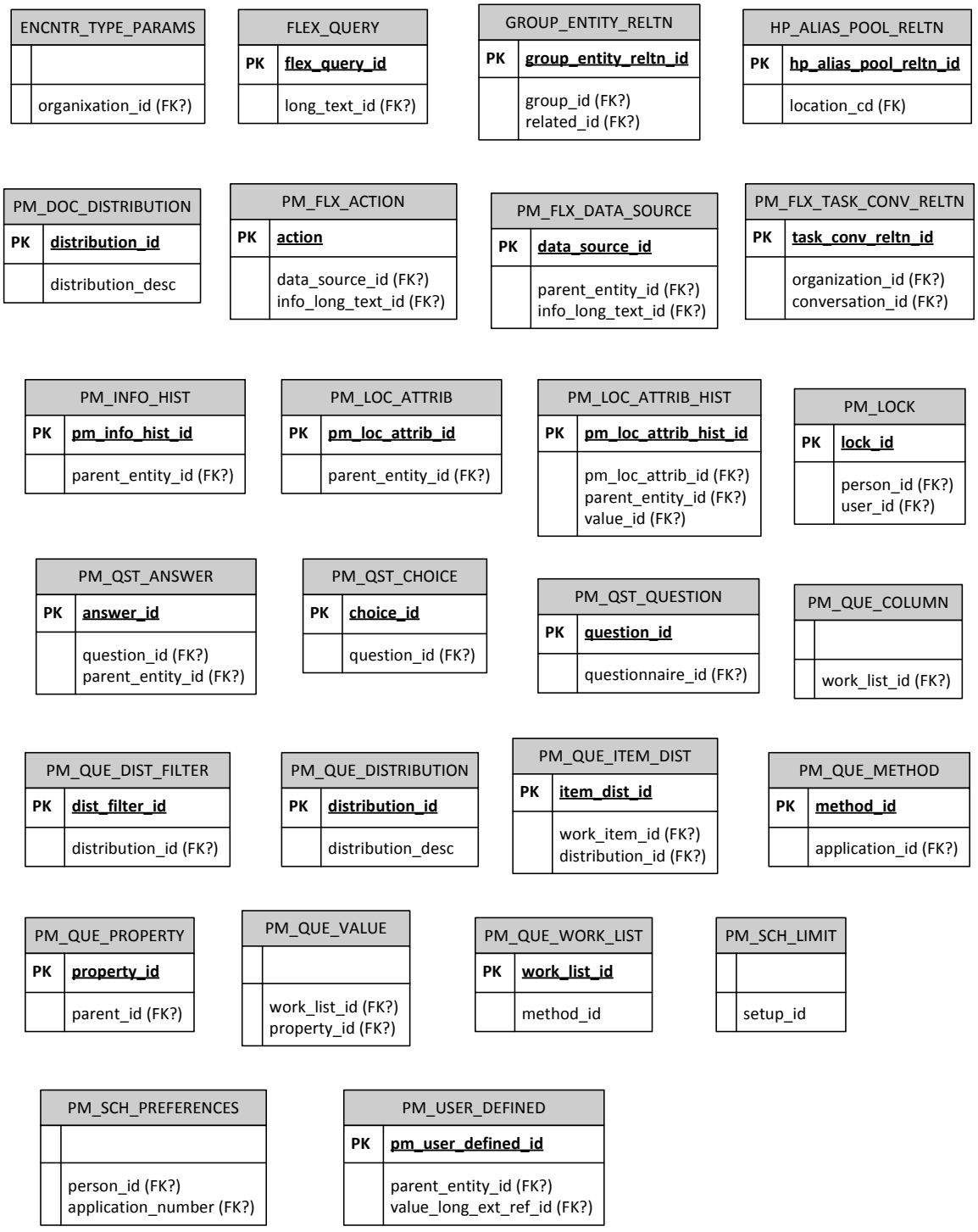
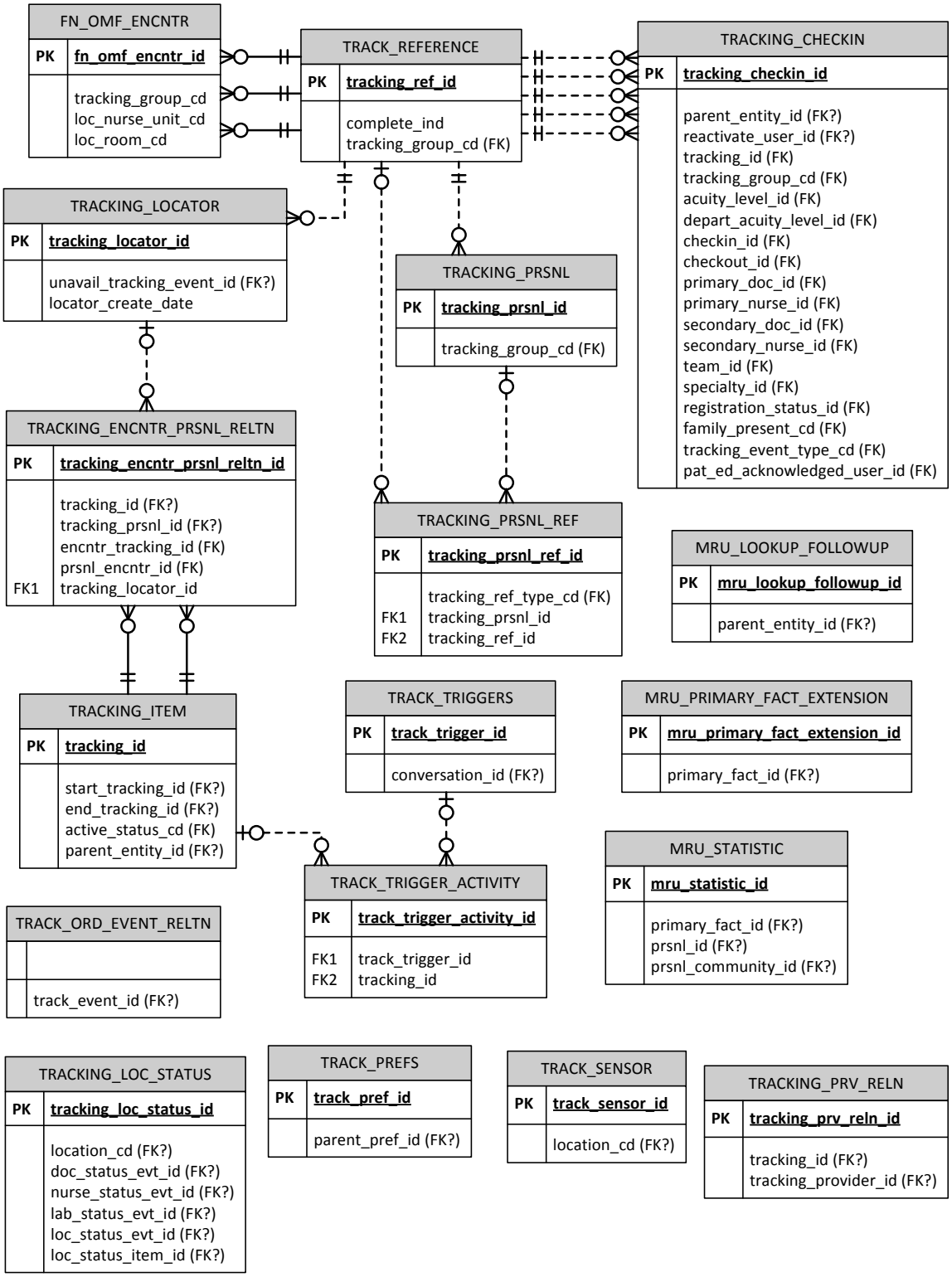


Figure 3. Redrawn ERD for Patient Tracking showing weaknesses in the ERD Design - Pt 1



Patient Tracking ERD Design Weaknesses - Pt 2

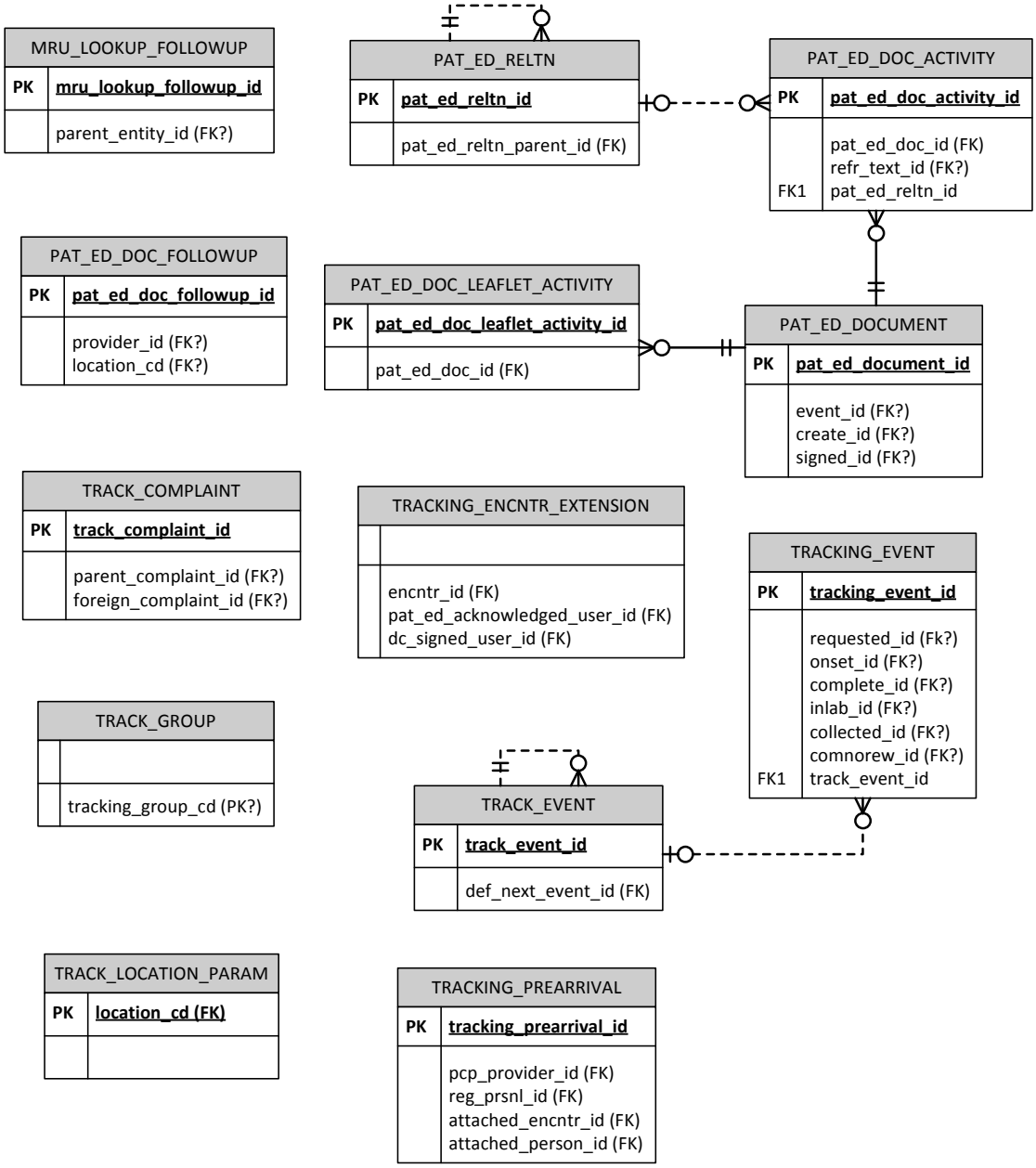


Figure 4. Redrawn ERD for Power Chart showing weaknesses in the ERD Design

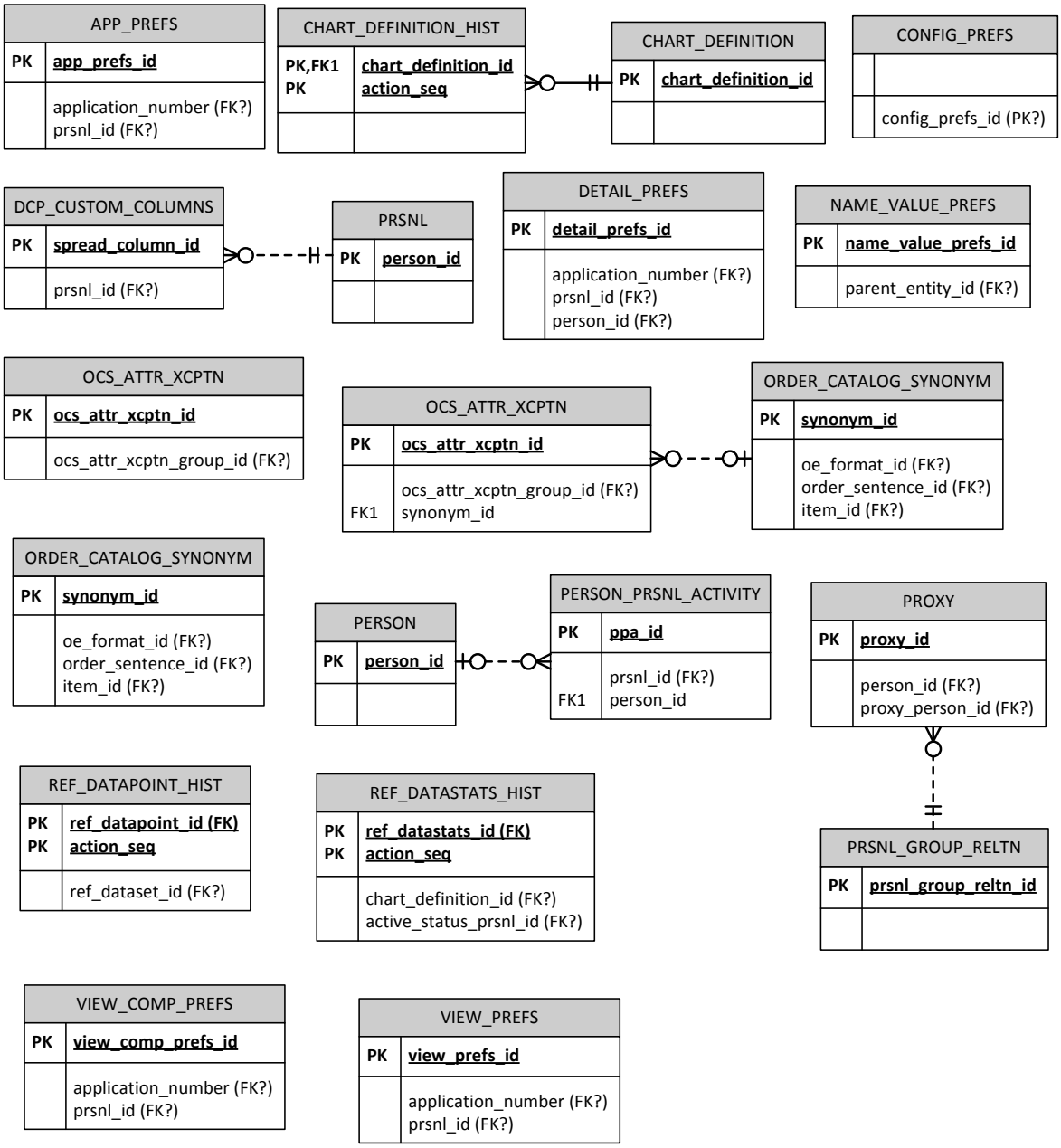


Figure 5. Redrawn ERD for the Security Application showing weaknesses in the ERD Design

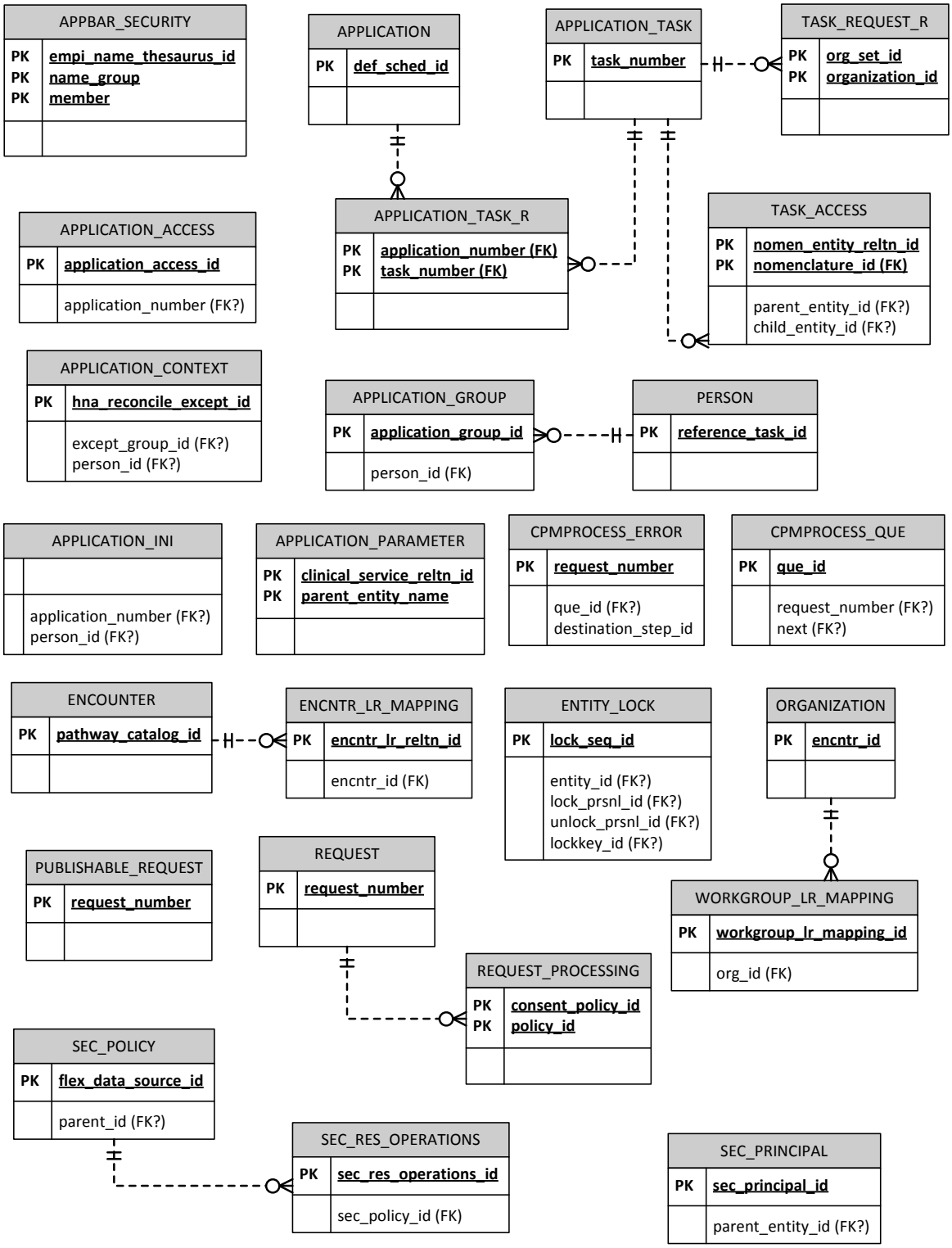
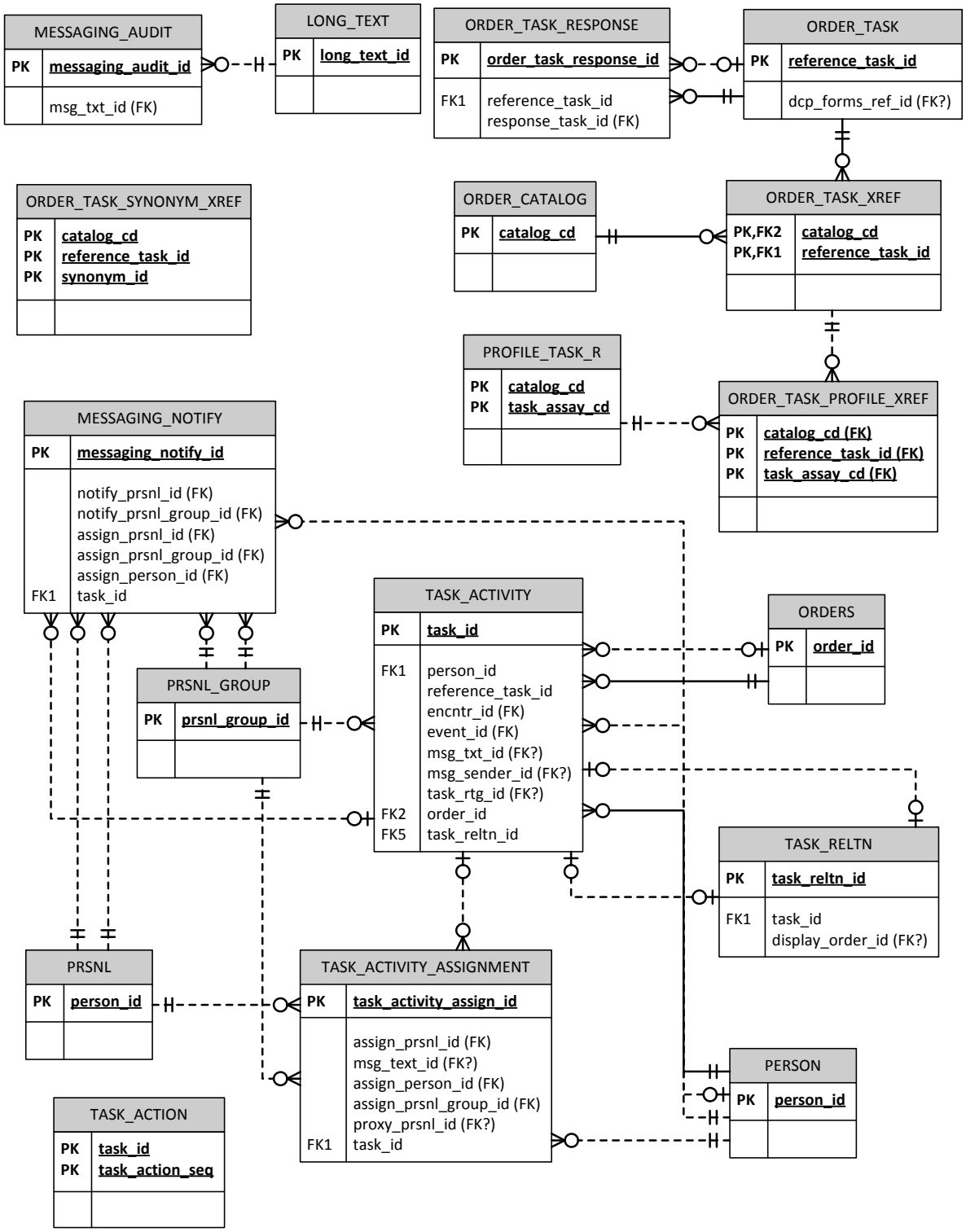



Figure 6. Redrawn ERD for the Task List showing weaknesses in the ERD Design