AD Applied Database Systems

Assignment 2. Due Friday, 12 November, 2010

Please staple your answers and put your NAME, Student ID and COURSE (e.g. MS.) on the front.
Unless the web site says otherwise (please check it) hand your answers into the Informatics Teaching Office, Appleton Tower by **16:00 Friday, 12 November.** Please consult the lab web page for instructions on how to turn in your project work.

Written Part - Assignment 2 [60pt]

1. A sequence of values or *time series* $v_0, v_1, v_2, \ldots$ is represented as a table $R(I, V)$ where I is the index or time and V is the value. For example, the sequence $3.4, 4.5, 5.7, \ldots$ is represented as the table

I	V
0	3.4
1	4.5
2	5.7
...	...

   (a) [3 points]Express in relational algebra the times at which the value is greater than 5.
   Answer: $\pi_I(\sigma_{V>5}(R))$ (In SQL: `SELECT I FROM R WHERE V > 5`)

   (b) Express SQL [3 points]and in relational algebra [4 points] the times at which the value is greater than the two neighbouring values, that is $v_{i-1} < v_i$ and $v_{i+1} < v_i$. *Hint.* For the relational algebra version construct a "next times" table. The table of all times and following times, e.g., $\{(0,1),(1,2),(2,3)\ldots\}$. This can be done by a query on the table.
   Answer:

```
SELECT R.I
FROM R, R1 AS R, R2 AS R
WHERE R.I = R1.I-1 AND R.I = R2.I+1 AND R1.V < R.V AND R2.V < R.V
```

   Relational algebra: First a table of unadjacent times (t_1, t_2) such that there is a third time t_3 with $t_1 < t_3 < t_2$ and to make things readable we simply construct a table of just the times $T = \pi_I(R)$:

$$U = \pi_{I_1 I_2}(\sigma_{I_1<I_2 \wedge I_3<I_2}(\rho_{I\to I_1}(T) \bowtie \rho_{I\to I_2}(T) \bowtie \rho_{I\to I_3}(T)))$$

   Second a table of "following" times:

$$F = \sigma_{I_1<I_2}(\rho_{I\to I_1}(T) \bowtie \rho_{I\to I_2}(T))$$

   So that the table of "next times" is $N = F \setminus U$
   Now we can get the desired result:

$$\pi_{I_2}(\sigma_{V_1<V_2 \wedge V_3<V_2}(\rho_{I\to I_1, V\to V_1}(R) \bowtie \rho_{I\to I_2, V\to V_2}(R) \bowtie \rho_{I\to I_3, V\to V_3}(R) \bowtie N \bowtie \rho_{I_1\to I_2, I_2\to I_3}(N)))$$

2. [10 points – 2 points each] For each of the following pairs of relational algebra expressions, say whether or not they are equivalent.

   - If they *are* equivalent, say which – in the absence of any knowledge about indexes – is probably the better evaluation plan;
   - if they are *not* equivalent, give an example (an instance of the tables involved) that shows they are not equivalent.

(a) $\sigma_{A=1}(R) - \sigma_{B=2}(R)$ and $\sigma_{(A=1)\wedge B\neq 2}(R)$
Answer: The same. $\sigma_{(A=1)\wedge B\neq 2}(R)$ is better – just one scan.

(b) $\pi_A(R) \cup \pi_A(S)$ and $\pi_A(R \cup S)$ where R and S are union-compatible.
Answer: The same. Eliminating duplicates from a union is costly. Probably better to project first and then do the union as there will be less i/o. I.e., $\pi_A(R) \cup \pi_A(S)$.

(c) $\pi_A(R) - \pi_A(S)$ and $\pi_A(R - S)$ where R and S are union-compatible.
Answer: Different. Consider $R = \begin{array}{c|c} A & B \\ \hline 1 & 2 \end{array}$ and $S = \begin{array}{c|c} A & B \\ \hline 1 & 3 \end{array}$

(d) $(R - S) \bowtie T$ and $(R \bowtie T) - (S \bowtie T)$ where R and S are union-compatible.
Answer: The same. Joins are potentially more costly than subtraction, so $(R - S) \bowtie T$.

(e) $\pi_A(R \bowtie S)$ and $\pi_A(R)$ where A is an attribute only of R.
Answer: Different. Consider $R = \begin{array}{c|c} A & B \\ \hline 1 & 2 \end{array}$ and $S = \begin{array}{c} B \\ \hline 3 \end{array}$

3. The IUPHAR receptor database is a widely used *curated database*. That is, it contains the work of a large number of experts who have extracted information from the scientific literature to produce a comprehensive and accurate reference work on pharmacological receptors. You can browse the database at `http://www.iuphar-db.org/GPCR/ReceptorFamiliesForward`. Do not be put off if you don't know any biology.

   The web pages you see are constructed on the fly from an underlying relational database, whose design is not entirely satisfactory. Your job is to design a better one.

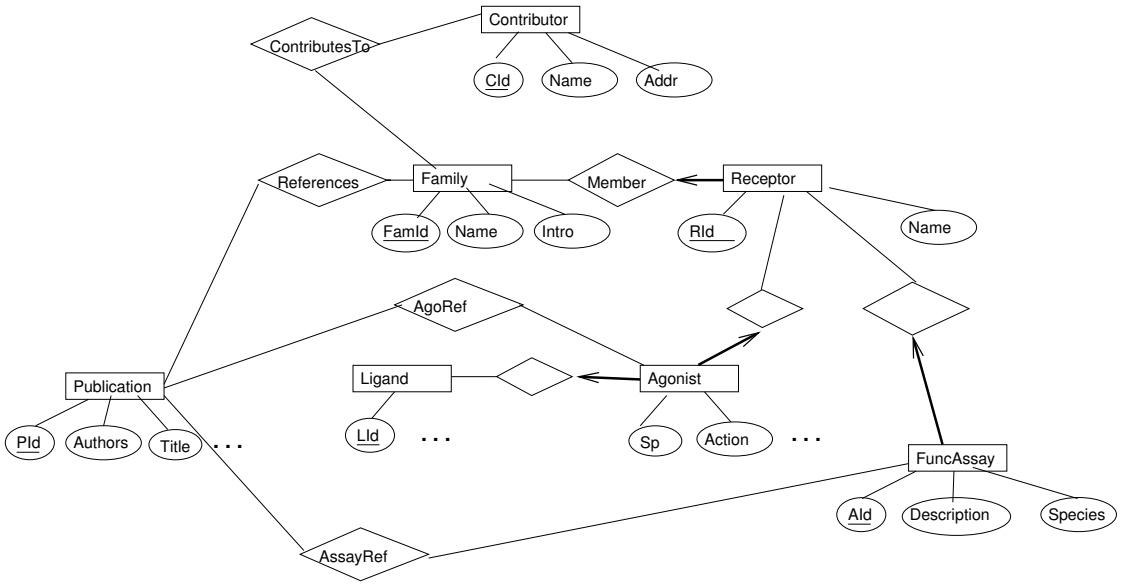
   To reduce the amount of "busy work", we ask you only to account for the following information in a receptor page: structural information, functional assays and agonists together with the information referenced by those tables. Your schema should describe receptor families and associated information.

   A few tips:

   - There should be an independent table or tables that describe publications.
   - In some places there is text with embedded references to publications. In a relational database it is not easy to ensure that those references are references to existing publications (a foreign key constraint). Try to come up with a partial answer to representing such a constraint
   - To see whether something has a key, it is sometimes useful to look at the URL

   You should hand in the following:

   (a) [5 points] an E-R diagram for the schema
   Answer: First, to address the issue of publications embedded in the text. This is something that XML (or rather XML with a schema or a DTD) could help us with, but this is a heavyweight solution. The best thing we can do is to write some software that scans the various text entries, checks that the citations (publication IDs) exist and generates numbers and hyperlinks for these.
   The design proposed here contains tables AgoRef and AssayRef that could be populated by this software.
   Now the E-R diagram:



Please see comments in the DDL for notes.

(b) [5 points]The SQL DDL, making sure to include all keys and foreign keys.
Answer:

```
CREATE TABLE Publication(
  PId INT,
  Authors TEXT,
  ...
  PRIMARY KEY PID)

CREATE TABLE Contributor(
  CId INT,
  Name TEXT,
  ...
  PRIMARY KEY CId)
  Note: Maybe not needed, but useful if a contributor contributes to more than one receptor

CREATE TABLE Family(
  FamId INT, Note: IUPHAR uses something called "chapter"
  Introduction: TEXT,
  PRIMARY KEY FamId)

CREATE TABLE References(
  FamId INT,
  PId INT,
  PRIMARY KEY (FamId,PId),
  FOREIGN KEY FamId REFERENCES Family,
  FOREIGN KEY PId REFERENCES Publication)

CREATE TABLE ContributesTo(
  CId INT,
  FamId INT,
  PRIMARY KEY (FamId,CId),
  FOREIGN KEY FamId REFERENCES Family,
```

```
  FOREIGN KEY CId REFERENCES Contributor)

CREATE TABLE Receptor(
  FamId INT,
  RId INT,
  Name TEXT,
  PRIMARY KEY (FamId,RId),
  FOREIGN KEY FamId REFERENCES Family)
```
Note: Not clear whether IUPHAR includes the Family key in the Receptor key
The receptor name might serve as a key, but it has complicated orthograpy.
An answer with an independent key is acceptable.

```
CREATE TABLE Ligand(
  LId INT,
  ...
PRIMARY KEY LId)

CREATE TABLE Agonist(
  FamId INT,
  RId INT,
  LId INT,
  Sp TEXT,
  Action TEXT,
  ...
  PRIMARY KEY (FamId, RId, LId),
  FOREIGN KEY LId REFERENCES Ligand,
  FOREIGN KEY (FamId, RId) REFERENCES Receptor(FamId, RId))

CREATE TABLE AgoRef(
  FamId INT,
  RId INT,
  PId INT,
  PRIMARY KEY (FamId, RId, PId),
  FOREIGN KEY PId REFERENCES Publication
  FOREIGN KEY (FamId, RId) REFERENCES Receptor(FamId, RId))

CREATE TABLE FuncAssay(
  AId INT,
  FamId INT,
  RId INT,
  Description TEXT
  Species TEXT
  PRIMARY KEY (FamId, RId, AId),
  FOREIGN KEY (FamId, RId) REFERENCES Receptor(FamId, RId))

  CREATE TABLE Assayref(
  FamId INT,
  RId INT,
  AId INT,
  PId INT,
```

PRIMARY KEY (FamId, RId, AId,PId),
FOREIGN KEY PId REFERENCES Publication
FOREIGN KEY (FamId, RId) REFERENCES Receptor(FamId, RId))

Note: You can put in ON DELETE CASCADE directives, if you think they are a good idea.

(c) For each of the following, the SQL query or queries that are needed to construct the following:

i. [3 points]A receptor family web page for a given the receptor family identifier.
Answer: The family web page as it appears in IUPHAR is very simple. It lists all the receptor families on the left-hand side. The names of the receptors in that family are needed. The names of the authors are also needed to generate the citation.
This is not required, but to get all the family names (and their IDs for generating web links):

SELECT FamId, Name
FROM Family

To get the receptor names and IDs, given a family ID N:

SELECT FamId, RId, Name
FROM Receptor
WHERE FamId = N

To get the contributor names:

SELECT C.Name
FROM Contributor C, ContributesTo CT
WHERE C.CId = CT.CId AND F.FamId = CT.FamId

ii. [3 points]The agonist table for a given receptor identifier.
Answer: The identifier will be a (FamId, RId) pair. Suppose they are f and r To get everything in the agonist table

SELECT FamId, RId, LId, Sp, Action, ...
FROM Agonist
WHERE FamId = f AND RId = r

To get the publications (we need the IDs to generate hyperlinks):

SELECT P.PId
FROM Publications P, AgoRef AR
WHERE P.PId = AR.PId AND AR.FamId = f AND Ar.RId = r

Of course, the web page generating software will have to build the non-1NF table and renumber the citations.

Note that you are not being asked to construct the web page or table, you are simply asked to produce a set of tables that could be used by some software to generate the web page with a minimal amount of extra work.

4. [6 points] – 3 for each correct Briefly describe *two* important ways in which relational algebra is more limited than SQL in what it can express.

Answer:

- SQL can do arithmetic
- SQL can operate on multisets as well as sets. This is sometimes useful.
- SQL has aggregate operations
- SQL has operators for update
- SQL allows null values, though the semantics is confusing,

5. (a) [3 points]Why is the following SQL DDL problematic?

```
CREATE TABLE Country(
   CountryName TEXT,
   Capital TEXT NOT NULL,
   PRIMARY KEY(Country)
   FOREIGN KEY(Capital) REFERENCES City(CityName) )

CREATE TABLE City(
   CityName TEXT,
   Country TEXT NOT NULL,
   PRIMARY KEY CityName,
   FOREIGN KEY(CountryName) REFERENCES COUNTRY(CountryName) )
```

Answer: One cannot construct a City tuple unless you already have a Country tuple and *vice versa*. So how does one populate the database?

(b) i. [3 points]Write down an SQL DDL declaration for entity sets A and B in which there is an ISA relationship between B and A (B isa A).
Answer:

```
CREATE TABLE A (
   AK INT,
   Astuff ...
   PRIMARY KEY (AK) )

CREATE TABLE B (
   AK INT,
   BStuff ...
   PRIMARY KEY (AK)
   FOREIGN KEY (AK) REFERENCES A(AK) )
```

ii. [3 points]Also write down the DDL in which B is a *weak entity set* dependent on A.
Answer:

```
CREATE TABLE A (
   AK INT,
   Astuff ...
   PRIMARY KEY (AK) )

CREATE TABLE B (
   AK INT,
   BK INT,
   BStuff ...
   PRIMARY KEY (AK,BK)
   FOREIGN KEY (AK) REFERENCES A(AK) )
```

Or one could have the BK alone be the key, in which case one would want to have NOT NULL for AK.

iii. [3 points]Describe the relationship bewteen the two declarations.
Answer: In the ISA relationship, the foreign key in B is exactly the primary key. In the weak entity relationship, the primary key may be part of the foreign key. In both cases a B entity depends on an A entity for its existence, and in both cases one can add an ON DELETE CASCASE if one wants dependent entities to be removed.

6. [6 points – 2 each] Consider the scheme:

Student(Id,Name,Email)	or	$S(\underline{I}, N, E)$
Takes(Id, Coursename, Marks)	or	$T(\underline{I}, \underline{C}, M)$
Assign(Coursename, Room, Lecturer, Period)	or	$A(\underline{C}, R, L, P)$

A "period" is a set of times at which courses can meet (e.g. MWF 11-12). Assume no two periods overlap.

(a) Write down the functional dependencies resulting from the key constraints on the S, T, and A tables.

Answer: $\{I \longrightarrow NE, IC \longrightarrow M, A \longrightarrow RLP\}$

(b) Write down a functional dependency that says that a student can only be in one place at a time.

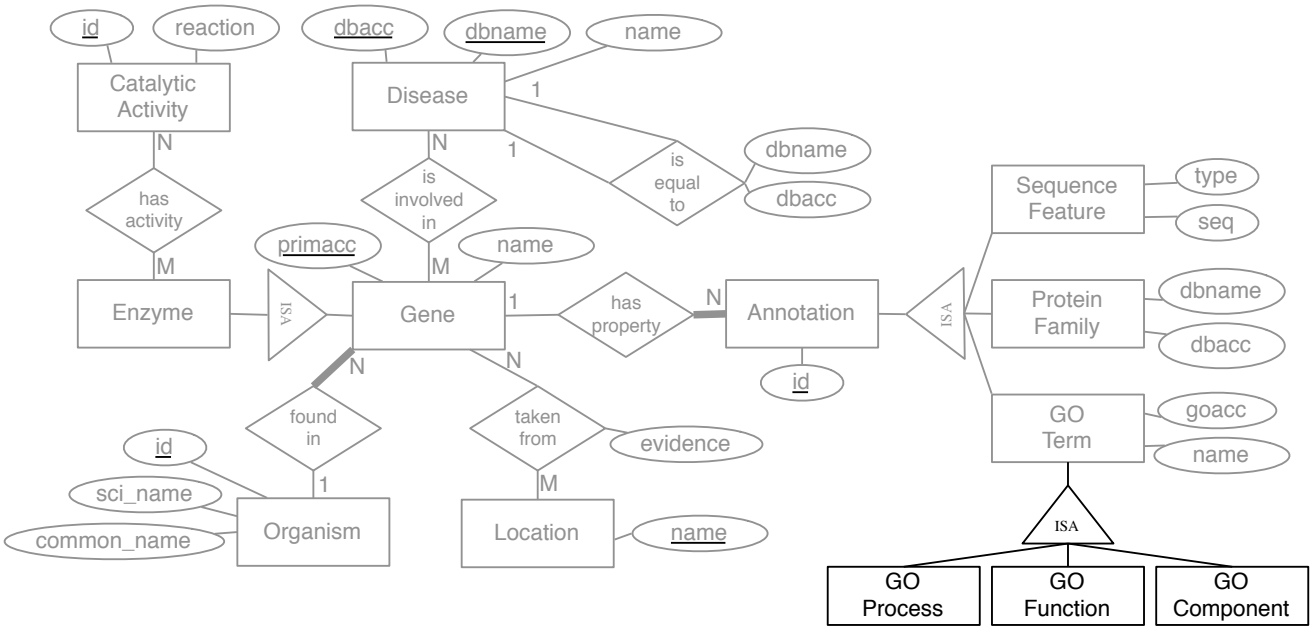
Answer: $IP \longrightarrow R$

(c) Write down a functional dependency that says that two different courses cannot be scheduled in the same room at the same time.

Answer: $RP \longrightarrow C$

Project Part - Assignment 2 [40pt]

While transforming the given ER schema into a relational schema during the first part of the project you noticed that the description about Gene Ontology terms contained more information than represented by the ER schema. Indeed, after talking to your colleagues you conclude that your database should be based on the following revised ER schema (note that only the highlighted part changed):



Your first task is to modify your relational database schema to conform to the revised ER schema (a common task in practical database design!). Note that we provide an example database schema for the previous assignment at

http://homepages.inf.ed.ac.uk/ntang/ad/schema.txt

You may use this example to revise your own solution before you start working on the solution for the current assignment.

After completing the database design process you can now start to populate your new database with data from the database that was provided by your partner institute (and which is now loaded into the DBMS

according to the old schema as part of the first assignment). You may need to perform additional changes to your database schema in order to ensure the following:

- Every tuple in relation `entry` results in a **Gene**, that is, results in a tuple in the relation representing the entity type **Gene**.
- There is a relation containing a list of unique (sub-cellular) location names. The list results from the attribute `text` in relation `comment` for those comments that are of type '*SUBCELLULAR LOCATION*'. The location names should be stripped from any of the following evidence suffixes: *(Potential)*, *(Probable)*, and *(By similarity)*. Further ensure that the trailing '.' of location names is omitted in your listing.
- When representing the relationship **taken from** between **Genes** and their **Location** (based on tuples in relation `comment`), the attribute `evidence` should take value '*Potential*', '*Probable*', and '*By similarity*' whenever corresponding evidence is given in the comment. In case that no evidence for a genes location is given the attribute `evidence` should be NULL.
- There is a relation containing a list of distinct occurrences of combinations of scientific and common names of organisms (in relation `entry`). Furthermore, these occurrences should be numbered starting from 1.
- Every **Gene** that corresponds to an `entry` having a comment of type '*CATALYTIC ACTIVITY*' results in an **Enzyme**.
- There is a list of distinct catalytic activities (resulting from comment texts of type '*CATALYTIC ACTIVITY*') numbered starting from 1.
- An **Enzyme** participates in relationship **has activity** with a **Catalytic Activity**, provided that there is a corresponding tuple in relation `comment`.
- List all annotations for a **Gene**: **Sequence Features** result from tuples in relation `feature`; **Protein Family** information results from database cross-references to databases other than '*GO*' or '*Orphanet*'; GO Terms result from database cross-references to '*GO*' (additional information on `name` and `type` is found in `go_term`).
- For sequence features the attribute `seq` contains only the substring of the gene sequence that is specified by the given `from_pos` and `to_pos` values in `feature`.
- Annotations to obsolete GO terms should not be included.
- **Disease** includes exactly one entry for each (i) distinct disease in '*Orphanet*' for which there is a database cross-reference, and (ii) every disease in `omim` (with `dbname` '*MIM*').
- Every **Disease** with `dbname` '*Orphanet*' for which there exists the same **Disease** with `dbname` '*MIM*' (based on equality of their `name`, ignoring character case) should participate in relationship **is equal to** (and vice versa).
- Participation of a **Gene** in relationship **is involved in** is derived from either a database cross-reference to '*Orphanet*', or from a `comment` of type '*DISEASE*' that contains a reference to an OMIM entry, e.g., "*Involved in susceptibility to Stevens-Johnson syndrome [MIM:608579].*" references entry *608579* in OMIM. In the latter case, you have to extract the reference from the text first. Assume that there exists only one such reference within each comment.

The following results are asked of you in this assignment:

1. A file containing the DDL statements for the modified database schema.
2. A file containing all the commands that are necessary in order to populate your database with data from the provided database.
3. A file containing queries (SQL statement and result) that count the number of corresponding tuples for each of the entity types in the revised ER schema and for relationships **taken from**, **has activity**, and **is involved in**.