BE768 Biological Databases Spring 2018

Exam

Name:

Problem 1: ER diagram

(35 points total)

Mutations, signaling pathways, and disease. There is much interest in prioritizing genetic mutations, discovered through sequencing experiments, based on their likely affect on human disease. One way that mutations can affect disease is through disruption of protein-protein interactions in signaling pathways. In these pathways, which transmit information from one part of a cell to another, proteins form complexes through interaction of their three-dimensional shapes. If a mutation disrupts a binding site on a protein, it can affect the ability of that protein to successfully bind to its interaction partners. This can then disrupt the ability of the signals to be successfully transmitted and may lead to disease. The goal of this question is to design a database which can be used for storing information to help prioritize mutational affects.

A. (25 points) Design and draw an **ER diagram** that captures the information below. Create entities and relations corresponding to the **bold face names** in the statements. Be sure to indicate:

- **the attributes for each entity (**and relation attributes if any**),**
- **keys for each entity and relation** (<u>underline</u> key fields),
- **relationship class** (one-to-many, many-to-many, etc.),
- **participation constraints** (total, partial),
- any **constraints that cannot be captured in the ER diagram**.

Use the symbols on the reference page in your diagram.

1. Each signaling **pathway** has an id, a name, and a function.
2. Each **protein** has an id, a name, a sequence, and a three-dimensional structure (stored as an external code from the protein data bank – PDB).
3. Proteins have pairwise **interact**ions. Each protein can have zero or more interactions. No more than one interaction can involve the same two proteins.
4. Each protein-protein interaction **participates in** exactly one signaling pathway or in none. Each pathway has at least one interaction.
5. A single nucleotide variation or **SNV**, has an id, a chromosome number and nucleotide position (where it occurs), a reference nucleotide, and a variant nucleotide.
6. A **binding site** has an id and a three-dimensional shape description (text).
7. Each protein **contains** zero or more binding sites. Each binding site is contained in exactly one protein.
8. An SNV can **alter** zero or more binding sites. Each binding site can be altered by one or more SNVs. When an SNV alters a binding site, a magnitude value is included to quantify the degree of alteration.
9. A **disease** has an id and a name.
10. A signaling pathway may be **involved** with zero or more diseases. Each disease is involved with at least one pathway.

B. (5 points) Write the create table statement for the **SNV** entity.

C. (5 points) Write the create table statement for the **interact** relationship. **Carefully** consider here your answers to part A. Assume that deletes in a foreign key table (if any) should cascade into this table but that updates should not. Write these assumptions explicitly. The format for create table statements is on the reference page.

Problem 2: Joins

(15 points total)

Consider the following two table instances:

R =

A	B	C
4	1	2
1	5	3
4	2	6
3	7	1

T =

B	C	D
2	1	5
3	8	2
4	7	1
2	6	4

Short answers:

1. (1 point) How many **columns** in R join T on R.B = T.B?

2. (1 point) How many **columns** in R join T using (B,C)?

3. (1 point) How many **rows** in Select * from R, T?

Give the resulting rows (label your columns).
Select *
4. (3 point) from R natural join T

5. (3 point) from R join T on R.A = T.D

6. (3 point) from R **left** join T using (C)

7. (3 point) from R **right** join T on R.A = T.B where A is NULL

Problem 3: SQL Select Statements

(35 points total, plus 5 bonus points)

Patient-derived cancer cell lines are studied to discover the genetic abnormalities that produce a cancer phenotype. You are given a cancer cell line database (below), which combines cell line information, gene information, gene mutations and gene expression.

Cell_Line (cid, name, age, sex, tissue, dataset)
Gene (gid, symbol)
Expression (*cid, gid*, fold_change)
Mutation (mid, *cid*, *gid*, chromosome, major_allele, variant_type)
(variant_type is one of "SNP", "INS", or "DEL")

Note: Primary keys are underlined. Field names that match in two tables are foreign keys (*italics*).

(35 points, 5 points each) Write SQL select statements for the following. You may use nested queries.

1. List all "colon" cancer cell lines (cid, name) that are "male" and at least 50 years of age in the "CCLE" dataset.

2. List all mutations and their gene symbols (mid, symbol) on chromosome 3 where the major allele is a "C". The list should be sorted in descending order of gene symbol.

3. List the genes (gid, symbol) that do NOT have any mutations.

4. List all cell lines (cid, name, tissue, count) that have at least 25 genes with a fold change greater than 2. The count is the number of genes.

5. List the cell lines (cid, name, tissue) that have:
   an expression fold change greater than or equal to 5 in the "TP53" genes,
   OR
   an expression fold change less than 4 in the "TNF" gene.
   Note "TP53" and "TNF" are gene symbols.

6. List all genes (gid, symbol) that NEVER have an expression fold change greater than 2 in any cell line. The list should be non-redundant. Be careful, this is a negative condition.

7. List all cell lines (cid) that have an expression fold change greater than 4 in the "BRCA1" gene and a "SNP" mutation in the "BRAF" gene.
   Note "BRCA1" and "BRAF" are gene symbols, and "SNP" is a variant type.

8. (Bonus 5 Points) List cell lines (cid, count) with the greatest number of recorded gene mutations. Count is the number of mutations. Careful, there may be ties.

Problem 4: Indexes

(10 points total)

1. (1 point) What type of index (clustered/unclustered) sorts the data?

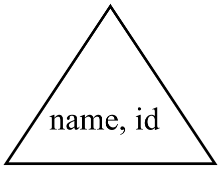
2. (1 point) If a table contains 30,000,000 pages of data and an index node can have 50 pointers, how many levels in the index (not counting the data pages)? Give the answer as a number or formula.

A database has the following tables:

Reviews (paperid, reviewerid, year)
Names (id, name)
Statistics (id, avg_rating)

The table **Reviews** stores reviewers for each manuscript submitted to a journal, **Names** stores reviewer names, **Statistics** stores the average rating of each reviewer (quality of the reviews). Ratings run from zero (poor) to 3 (excellent). You want to execute a query (below) that returns the names of reviewers in year 2012 who had an average rating greater than 2.5.

The only table that has an index is **Names** (figure below). It has a primary index on (name, id). In **Statistics**, id is unique, and avg_rating is not. In **Reviews**, no single field is unique but any pair with reviewerid is unique. In **Names**, no single field is unique, but the pair is unique.

			Index
Reviews	Statistics	Names	Table
8000	5000	5000	Number of records

The explain for the query returns the following table:

```
mysql> explain
    -> select id, name, avg_rating
    -> from Names natural join Statistics join Reviews on reviewerid=id
    -> where avg_rating > 2.5 and year = 2012;
+----+-------------+-------------+------+---------------+------+---------+------+------+----------+------------+
| id | select_type | table       | type | possible_keys | key  | key_len | ref  | rows | filtered | Extra      |
+----+-------------+-------------+------+---------------+------+---------+------+------+----------+------------+
|  1 | SIMPLE      | Reviews     | ALL  | NULL          | NULL | NULL    | NULL | 8000 |    10.00 | Using where|
|  1 | SIMPLE      | Statistics  | ALL  | NULL          | NULL | NULL    | NULL | 5000 |     3.33 | Using where|
|  1 | SIMPLE      | Names       | ALL  | NULL          | NULL | NULL    | NULL | 5000 |    10.00 | Using where|
+----+-------------+-------------+------+---------------+------+---------+------+------+----------+------------+
3 rows in set, 1 warning (0.00 sec)
```

3. (1 point) Can reviewerid be used as a primary key in **Reviews**?

4. (1 point) About how many record combinations are produced by this query as indicated by the explain?

5. (3 points) Suppose as in the explain, that a record is pulled from **Reviews** first. Suggest an index for **Reviews** that can be used to efficiently retrieve all the records with year = 2012. Efficiently means that not all records in reviews need to be examined. **Your answer should include:** the type of index (primary, index, unique), the field(s) of the index in order, and the alter table statement for your index. The syntax is on the reference page.

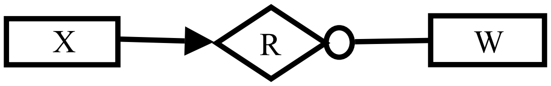
6. (3 points) For a record (below) taken from **Reviews**, suggest an index for **Names** that can be used to efficiently retrieve the correct record matching the reviewerid. **Your answer should include:** the type of index (primary, index, unique), the field(s) of the index in order, and the alter table statement for your index. You cannot change the existing index in **Names**.

paperid	reviewerid	year
300	400	2012

Problem 5: Short Answers

(2 points each; 6 points total)

1. For the relation R below, do we create a new table?



2. Can an aggregate function (count, max, min, etc.) be used in a WHERE clause? Why or why not? No more than one sentence.

3. Suppose you have the following tables, with B.aid a foreign key referencing A.aid:
   A(aid, …)
   B(bid, aid, …)
   Explain how the contents of A can prevent insertion of a record into B.
   No more than two sentences.

Reference Page

Create Table Format

```
CREATE TABLE tbl_name (
   col_name data_type [NOT NULL | NULL][DEFAULT default_value] [AUTO_INCREMENT],
   PRIMARY KEY (index_col_name,...),
   FOREIGN KEY (index_col_name,...) REFERENCES tbl_name (index_col_name,...)
 [ON DELETE {CASCADE | SET NULL | NO ACTION | SET DEFAULT}]
 [ON UPDATE {CASCADE | SET NULL | NO ACTION | SET DEFAULT}]
) ENGINE = INNODB
```

data type:

```
INT[(length)]
REAL[(length,decimals)]
DOUBLE[(length,decimals)] [UNSIGNED] [ZEROFILL]
FLOAT[(length,decimals)] [UNSIGNED] [ZEROFILL]
DECIMAL(length,decimals) [UNSIGNED] [ZEROFILL]
CHAR(length) [BINARY | ASCII | UNICODE]
VARCHAR(length) [BINARY]
DATE
TIME
TEXT
ENUM(value1,value2,value3,...)
```

Select Statement Format

```
SELECT select_expression
[FROM table_references]
[WHERE where_definition]
[GROUP BY col_list]
[HAVING having_definition]
[ORDER BY col_list [ASC | DESC]]
[LIMIT row_count]
```

Create Index Format

```
ALTER TABLE table_name
ADD PRIMARY KEY (field_name,…)

ALTER TABLE table_name
ADD [UNIQUE|INDEX] index_name (field_name,…)
```

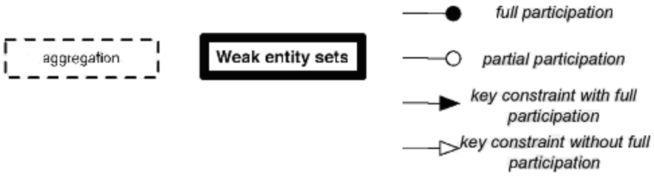
Using LIKE:
LIKE matches the entire string
% means zero or more characters
_ (underscore) means any one character

examples (TRUE if):
LIKE '%pat%' – pat somewhere in string
LIKE 'pat%' – pat at beginning of string
LIKE '_pat%' – pat starting at second character

Using REGEXP:
REGEXP matches anywhere in the string

example (TRUE if):
REGEXP 'pat1|pat2|pat3' – pat1 or pat2 or pat3 anywhere in string

ER Diagram Legend


aggregation
Weak entity sets
full participation
partial participation
key constraint with full participation
key constraint without full participation