BE768 Biological Databases Spring 2019

Exam

Name:

Problem 1: ER diagram

(35 points total)

Clinical Trials. When potential new drugs are developed by pharmaceutical companies, they must undergo clinical trials to test for safety, effectiveness, and the ability to improve outcomes over existing treatments. Clinical trials involve patients who take the drugs and doctors who supervise the trials. The phase of a clinical trial (e.g., I, II, III) indicates the information goal of the trial. For example, phase I trials test primarily for safety. The goal of this question is to design a database to store information about clinical trials, drugs, and participants.

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** (underline 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.

Note that ternary or aggregate relationships are not necessary for this ER diagram.

1. A **disease** has an id, a name, a type classification (from an enumerated list), and a prevalence (proportion in the population).
2. A **drug** has an id, a name, and a formula.
3. A drug **treats** a disease. A drug may treat more than one disease and a disease may have zero or more drugs that treat it.
4. A **patient** has an id, a biological gender, an ethnicity, and an age at the start of the trial.
5. A **doctor** has an id, a name, a specialization, and an affiliated medical institute.
6. A **clinical trial** has an id, a start date, an end date, a phase (enumerated I,II,III), and a regulatory agency approval status (a character string).
7. A clinical trial **evaluates** a drug. Each drug can be evaluated in zero or more clinical trials. Each clinical trial evaluates exactly one drug.
8. Patients **participate in** clinical trials. Each patient participates in exactly one clinical trial and receives a drug dosage. Each clinical trial has several patients participating.
9. A medical monitoring **test** has an id, and a name.
10. All patients **undergo** one or more monitoring tests and for each test, the date and result is recorded. Each test is used for zero or more patients.
11. Each clinical trial is **supervised by** exactly one doctor. A doctor supervises one or more clinical trials.

B. (5 points) Write the create table statement for the **disease** entity. For the type classification, assume the types are labeled T1, T2, T3, T4, T5. The format for create table statements is on the reference page.

C. (5 points) Write the create table statement for the **clinical trial** entity. **Carefully** consider here your answers to part A. Assume that updates in a foreign key table (if any) should cascade into this table but that deletes should not. Write these assumptions explicitly.

Problem 2: Joins

(15 points total)

Consider the following two table instances:

R =

A	B	C
8	1	5
1	3	3
8	2	7
4	9	2

T =

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

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 natural join T?

3. (1 point) How many c**olumns** in R join T using (C)?

Give the resulting rows (label your columns).

Select *

4. (3 point) from R join T using (B,C)

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

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

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

Problem 3: SQL Select Statements

(30 points total, plus 5 bonus points)

Observed genetic variation can be used to help understand phenotype and disease outcome. Several types of large-scale variation are termed structural variations (SVs). These include copy number variation (CNV – a large part of a chromosome is duplicated or lost) and inversion (INV -- a large part is inverted). You are given a human structural variation database with information on individual structural variants and genomes where they have been validated.

SV(sid, type, chr, start, end) **Note**: type is one of (CNV, INV)
genome(nid, gname, *popid*)
validation(*nid, sid*)
population(popid, popname)

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

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

1. List all SVs that start on chromosome 4 between indices 100 million and 150 million. Order reverse alphabetically by type and then by increasing start position. (sid, type, start)

2. List all 'INV' SVs validated in the genome with gname = 'NA12878'. (sid, chr, start, end)

3. List all genomes with no SVs validated. (nid, gname)

4. For each population, list the number of genomes in the database. Report only those populations with at least 3 genomes. (popname, count)

5. List the SV(s) with the numerically lowest ending index on chromosome 7. Note there may be more than one. (sid, type, end)

6. List any SV that has been validated in at least two members of the same population. (sid, popname)

7. Give the average number of validated SVs across all the genomes in the validation table. (single number answer: average)

Problem 4: Indexes

(10 points total)

A database has the following tables:

Reviewers2019 (paperid, reviewerid, year)
Name (id, name)
Email (id, email)
Statistics (id, avg_rating, return_to_request_ratio)

The table **Reviewers2019** stores reviewers for each manuscript submitted to a journal this year. **Name** stores names, **Email** stores emails for people listed in Name. **Statistics** stores information about reviewer performance in the past (average rating, percentage of times returned a review). Ratings run from 0 (poor) to 3 (excellent). Return to request ratio is a number between 0 and 1. You want to execute a query (below) that returns the names of reviewers who had an average rating of at least 2 and a return ratio of at least 0.8.

The only table that has an index is **Name** (figure below). It has a primary index on (id). In **Email**, neither id nor email is unique, but the combination is unique. In **Statistics**, id is unique, but other fields are not. In **Reviewers2019**, no single field is unique but a combination of paperid and reviewerid is unique.

Index

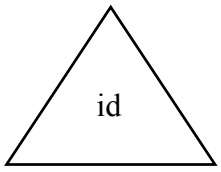


Table	Name	Email	Statistics	Reviewers2019
Rows	5000	6000	5000	300

The explain for the query returns the following table:

```
mysql> explain
-> select *
-> from Name join Email using(id) join Statistics using (id)
-> join Reviewers2019 on reviewerid=id
-> where rating>=2 and return_to_request_ratio >= 0.8;
+----+-------------+---------------+--------+---------------+---------+---------+------------+------+-------------+
| id | select_type | table         | type   | possible_keys | key     | key_len | ref        | rows | Extra       |
+----+-------------+---------------+--------+---------------+---------+---------+------------+------+-------------+
|  1 | SIMPLE      | Reviewers2019 | ALL    | NULL          | NULL    | NULL    | NULL       |  300 | NULL        |
|  1 | SIMPLE      | Name          | eq_ref | PRIMARY       | PRIMARY | 4       | reviewerid |    1 | NULL        |
|  1 | SIMPLE      | Statistics    | ALL    | NULL          | NULL    | NULL    | NULL       | 5000 | Using where;|
|  1 | SIMPLE      | Email         | ALL    | NULL          | NULL    | NULL    | NULL       | 6000 | Using where;|
+----+-------------+---------------+--------+---------------+---------+---------+------------+------+-------------+
4 rows in set, 1 warning (0.00 sec)
```

1. (2 points) What is the depth of a clustered index search tree (number of layers **above** the data pages) for 10,000,000 records if the fan out (number of pointers in a tree node) is 40? **Give a formula.**

2. (1 point) If you add another index to **Name**, will it be clustered or unclustered?

3. (1 point) About how many record combinations are produced by this query as indicated by the explain?

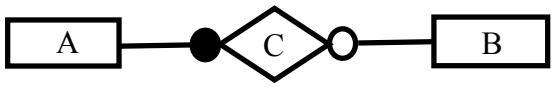
4. (3 points) The explain shows that a record is pulled from **Reviewers2019** first (mainly because it's the smallest table), and then the primary key index is used in **Name** to find matching records. Suggest an index for the next table, **Statistics,** that can be used to efficiently retrieve matching records. Efficiently means that not all records in **Statistics** need to be examined. **Your answer should include:** the type of index (primary key, index, unique), the field(s) of the index in order, the alter table statement for your index, and a one-sentence justification for your answer. The syntax is on the reference page.

5. (3 points) Suggest an index for the next table in the explain output, **Email,** that can be used to efficiently retrieve matching records. **Your answer should include:** the type of index (primary key, index, unique), the field(s) of the index in order, the alter table statement for your index, and a one-sentence justification for your answer.

Problem 5: Short Answers

(2 points each; 10 points total)

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



2. True or False. Every primary key field must be declared "not null".

3. True or False. Every foreign key field must be declared "not null".

4. How do you indicate that an aggregate function (count, max, min, avg, etc.) should be used on an entire table rather than a sub-table? No more than one sentence.

5. Suppose you have the following tables, with T.sid a foreign key referencing S.sid:
   S(sid, …)
   T(tid, …, sid)
   Explain how T can block deletion of a record in S.
   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]

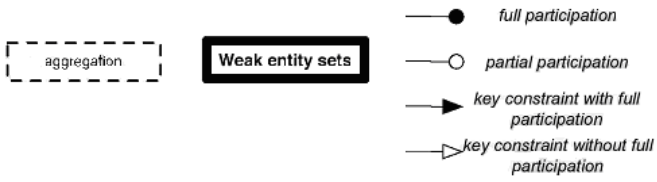
Using LIKE:
LIKE matches the entire string
% means zero or more characters
_ (underscore) means any one character

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



Create Index Format

ALTER TABLE table_name
ADD PRIMARY KEY (field_name,…)

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