COMS W4111-003 (Fall 2022)
Introduction to Databases

Homework 2: Non-Programming and Non-Programming

Note: Please replace the information below with your last name, first name and UNI.

Wu, Tong, tw2906

Setup Environment

In [5]:

```
%load_ext sql
```

```
The sql extension is already loaded. To reload it, use:
  %reload_ext sql
```

In [6]:

```
%sql mysql+pymysql://root:dbuserdbuser@localhost
```

In [7]:

```
%sql select * from db_book.classroom
```

```
 * mysql+pymysql://root:***@localhost
5 rows affected.
```

Out[7]:

building	room_number	capacity
Packard	101	500
Painter	514	10
Taylor	3128	70
Watson	100	30
Watson	120	50

Introduction

Structure

This homework has four sections:

- PART A: Written questions on concepts covered in class.

- PART B: Common problems for both the programming and non-programmiong tracks.

Because of delays in progress and lectures, I am not defining track specific questions for HW 2.

Submission

Please refer Ed Discussion Announcement for the submission instructions.

This assignment is due October 30, 11:59 pm EDT

Collaboration

- You may use any information you get in TA or Prof. Ferguson's office hours, from lectures or from recitations.
- You may use information that you find on the web.
- You are NOT allowed to collaborate with other students outside of office hours.

Part A: Written

Place your answers in the Markdhown cells following each question. Your answers should be succinct. We will deduct points for long or rambling answers.

W1

Question:

Codd's 3rd Rule states: "Null values (distinct from the empty character string or a string of blank characters and distinct from zero or any other number) are supported in fully relational DBMS for representing missing information and inapplicable information in a systematic way, independent of data type."

Briefly explain the meaning of this rule.

The example database from the book has a table `takes` with a column `grade`. The value is `NULL` if a student took the course but did not get a grade. Why would using the string "NA" instead of `NULL` cause problems for some queries on this table?

Answer:

The 3rd rule means that, the NULL value is a state but not a actual value to determine the number of the specific object or attribute. The NULL means the system do not know and do not have the actual number of that object. NULL value is not zero value since the zero means that the database know the value of the object.

The NULL value in takes database means the student took the course but not get a grade. Since the grade is also showed as sting, like 'A-' and 'C+', using NULL value can easy to distinguish the grade from 'string' type and NULL type. So the example of the effected situation may that, a query to find that the student take this class but not get grade, use 'grade is NOT NULL' to query, if the 'NA' is inserted instead of NULL, then the query will not find the correct answer, and vice versa.

W2

Question:

Codd's 4th Rule states: "The data base description is represented at the logical level in the same way as ordinary data, so that authorized users can apply the same relational language to its interrogation as they apply to the regular data."

- What is the schema that contains the database description information for MySQL?
- Give three examples of information about database structure that is in the schema.

Answer:

The metadata is the data of the data in the database, it need to be shown to users as a same way of the database table to allow users to know whether the table or the database that he/she query is exist or not. The schema that contains the metadata of the database, also called description, is called information_schema.

The table 'TABLES' stored all tables in all databases. The table 'COLUMNS' stored all columns that each database and each table has. The table '. And the table 'KEY_COLUMN_USAGE' stored all keys that tables are used.

W3

Question:

- What is the primary reason for creating indexes?
- Why is creating very many indexes potentially a problem? What is the negative affect of creating unecessary indexes?

Answer:

Index allows the system to find tuples in relation that have a specific value for that attribute in a relative high efficiency, compare to the inefficient scanning all the records.

Many indexes will cost much more storage in the disk, since each index is a copy of the data so it need more index. Also if there are many unecessary indexes, it will cause that if a value need to be inserted into a table, all indexes that related to this attributed need to be update, so it will slows down the overall write speed of the database.

W4

Question:

- What is the primary reason for creating indexes?
- Why is creating very many indexes potentially a problem? What is the negative affect of creating unecessary indexes?

Answer:

Index allows the system to find tuples in relation that have a specific value for that attribute in a relative high efficiency, compare to the inefficient scanning all the records.

Many indexes will cost much more storage in the disk, since each index is a copy of the data so it need more index. Also if there are many unecessary indexes, it will cause that if a value need to be inserted into a table, all indexes that related to this attributed need to be update, so it will slows down the overall write speed of the database.

W5

Question:

- In SQL, what is the main difference between a primary key and a unique key?

Answer:

Both guarntee the uniqueness of the column. A table can only has one primary key, but can has multiple unique key.

W6

Question:

- Views are a valuable concept in relational databases. What are three distinct reasons for/benefits of creating views?

Answer:

1. Views allow to hide some certain data for users, it limit the access of users to database, probide more security.
2. Views can be formed from multiple tables, it provides a easier querying.
3. Views can provide integrity constraints since it will check integrity constraints when user insert or edit data from views.

W7

Question:

- Explain the concept of a *domain?* for table column values.
- Consider Columbia Course numbers, e.g. W4111, E1006, C1001. What is the domain for course numbers not just `CHAR(5)`.

Answer:

Domain provides a data type which can has constrains, it defines the type and rules to follow for a column of values . For the columbia course numbers, it should have a domain to constrain that char(5) fromed by one letter and four digits.

W8

Question:

- List two examples of `integrity constraints` that apply to a single table, and one example that applies to multiple tables.

Answer:

Apply to a single table:
not null, unique

Apply to multiple tables: foreign key

W9

Question:

Consider the table `time_slot` from the sample database associated with the recommended text book.

- The data type for the column `day` is `char(1)`. Given the data types MySQL supports, what is a better data type?
- What is a scenario that would motivate creating an index on the column `day` ?

Answer:

Store with DATE data type and use function weekday() to get the week day. If there has many columns and rows in the table time_slot, and user need to query the data by using 'day'. In this case, creating an index for column 'day' is useful.

W10

Question:

Consider the table `course` from the sample database associated with the recommended text book.

- There is a design problem with the column `course_id` . What is the problem and how would you fix it?

Answer:

The column course_id also used in table section, which is a weak entity related to course since it use course_id as foreign key and cannot form a uniqueness without course_id. It can be seen as a detailed table of course table.

Part B: Common Tasks

- You will use the example datatabase from the book associated with the class and the Lahman baseball data you loaded from HW 0 to answer these questions.
- Execute the SQL you write as answers in the answer cells.

C 1

Question:

Write a query that produces the following table. You must match column names and formatting of values.

dept_name	no_of_courses	budget	cost_per_course
Biology	2	90000.00	45000.0
Comp. Sci.	8	100000.00	12500.0
Elec. Eng.	1	85000.00	85000.0
Finance	1	120000.00	120000.0
History	1	50000.00	50000.0
Music	1	80000.00	80000.0
Physics	1	70000.00	70000.0

Answer:

In [48]:

```
%%sql
use db_book;
SELECT
    course.dept_name,
    count(course.course_id) AS no_of_courses,
    department.budget,
    convert(department.budget/count(course.course_id), decimal(10,1)) as cost_per_cost
FROM db_book.course
NATURAL JOIN
    department
GROUP BY dept_name;
```

```
 * mysql+pymysql://root:***@localhost
0 rows affected.
7 rows affected.
```

Out[48]:

dept_name	no_of_courses	budget	cost_per_cost
Biology	3	90000.00	30000.0
Comp. Sci.	5	100000.00	20000.0
Elec. Eng.	1	85000.00	85000.0
Finance	1	120000.00	120000.0
History	1	50000.00	50000.0
Music	1	80000.00	80000.0
Physics	1	70000.00	70000.0

C 2

Question

- Use the `people` table for Lahman Baseball data for this query.
- Write a query that produces a result with the following columns:
  - `first_initial` is the first first letter of `nameFirst` followed by .
  - `nameLast`
  - `place_of_birth` is the `birthCity` , a comma, and the `birthCountry`.
- You can just run your queries for the first 10 people (fyi, the example table below have more than the first 10 people)

initial	nameLast	place_of_birth
D.	Aardsma	CO, USA
H.	Aaron	AL, USA
T.	Aaron	AL, USA
D.	Aase	CA, USA
A.	Abad	FL, USA
F.	Abad	La Romana, D.R.
J.	Abadie	PA, USA
E.	Abbaticchio	PA, USA
B.	Abbey	VT, USA
C.	Abbey	NE, USA
D.	Abbott	OH, USA
F.	Abbott	OH, USA
G.	Abbott	AR, USA
J.	Abbott	GA, USA
J.	Abbott	MI, USA
K.	Abbott	OH, USA
K.	Abbott	MA, USA

O.	Abbott	PA, USA
P.	Abbott	CA, USA
A.	Aber	OH, USA

Answer

In [49]:

```
%%sql
use lahmansbaseballdb;
SELECT
    concat(left(people.nameFirst, 1), '.') as initial,
    nameLast,
    concat(birthState, ', ', birthCountry) as place_of_birth
FROM people
LIMIT 10
```

```
 * mysql+pymysql://root:***@localhost
0 rows affected.
10 rows affected.
```

Out[49]:

initial	nameLast	place_of_birth
D.	Aardsma	CO, USA
H.	Aaron	AL, USA
T.	Aaron	AL, USA
D.	Aase	CA, USA
A.	Abad	FL, USA
F.	Abad	La Romana, D.R.
J.	Abadie	PA, USA
E.	Abbaticchio	PA, USA
B.	Abbey	VT, USA
C.	Abbey	NE, USA

C3

Question

- Use the tables `people`, `appearances`, `batting` from the Lahman's Baseball data to answer this question.
- Produce a table of the form:
  - `playerID`

- `nameLast`
- `nameFirst`
- `career_teams` is a semi-colon separated list of the team
- `career_games` is the sum of `G_all` from `appearances`
- `total_abs` is the sum of `AB` from `batting`
- `total_hits` is the sum of `h` from `batting`
- `batting_avg` is `total_hits/total_abs` is `total_abs` is not 0, and is ```NULL`` otherwise.

- Show the first 10 rows like below.

playerID	nameLast	nameFirst	career_teams	career_games	total_abs	total_hits	batting_avg
aardsda01	Aardsma	David	ATL;BOS;CHA;CHN;NYA;NYN;SEA;SFN	331	4	0	0.0000
aaronha01	Aaron	Hank	ATL;ML1;ML4	3298	12364	3771	0.3050
aaronto01	Aaron	Tommie	ATL;ML1	437	944	216	0.2288
aasedo01	Aase	Don	BAL;BOS;CAL;LAN;NYN	448	5	0	0.0000
abadan01	Abad	Andy	BOS;CIN;OAK	15	21	2	0.0952
abadfe01	Abad	Fernando	BOS;HOU;MIN;OAK;SFN;WAS	384	9	1	0.1111
abadijo01	Abadie	John	BR2;PH3	12	49	11	0.2245
abbated01	Abbaticchio	Ed	BSN;PHI;PIT	857	3044	772	0.2536
abbeybe01	Abbey	Bert	BRO;CHN;WAS	79	225	38	0.1689
abbeych01	Abbey	Charlie	WAS	452	1756	493	0.2808

Answer 1

In [11]:

```
%%sql
use lahmansbaseballdb;
drop table c3;
CREATE TABLE c3 AS
    SELECT
        people.playerID,
        people.nameLast,
        people.nameFirst,
        replace(GROUP_CONCAT(DISTINCT appearances.teamID), ',', ';') as career_teams,
        sum(DISTINCT appearances.G_all) as career_games,
        sum(DISTINCT batting.AB) as total_abs,
        sum(DISTINCT batting.H) as total_hits,
        convert(sum(DISTINCT batting.H)/sum(DISTINCT batting.AB), decimal(10,4)) as batting_avg
    FROM
        people NATURAL JOIN
            appearances
        JOIN
            batting on
                batting.playerID = people.playerID
    GROUP BY
        people.playerID;
```

```
 * mysql+pymysql://root:***@localhost
0 rows affected.
0 rows affected.
19689 rows affected.
```

Out[11]:

```
[]
```

Answer 2

Demonstrate that you computed `batting_avg` correctly by returning the the first 10 rows with a null batting average like below.

playerID	nameLast	nameFirst	career_teams	career_games	total_abs	total_hits	batting_avg
abbotgl01	Abbott	Glenn	DET;OAK;SEA	248	0	0	None
abreubr01	Abreu	Bryan	HOU	7	0	0	None
abreuju01	Abreu	Juan	HOU	7	0	0	None
achteaj01	Achter	A. J.	LAA;MIN	45	0	0	None
acrema01	Acre	Mark	OAK	114	0	0	None
adamja01	Adam	Jason	KCA;TOR	54	0	0	None
adamsch01	Adams	Chance	NYA	16	0	0	None
adamswi02	Adams	Willie	OAK	25	0	0	None
adenhni01	Adenhart	Nick	LAA	4	0	0	None
adkinst01	Adkins	Steve	NYA	5	0	0	None

In [16]:

```
%%sql
use lahmansbaseballdb;
SELECT
    * from c3
    WHERE batting_avg is NULL
    LIMIT 10
```

```
 * mysql+pymysql://root:***@localhost
0 rows affected.
10 rows affected.
```

Out[16]:

playerID	nameLast	nameFirst	career_teams	career_games	total_abs	total_hits	batting_avg
abbotgl01	Abbott	Glenn	DET;OAK;SEA	229	0	0	None
abreubr01	Abreu	Bryan	HOU	7	0	0	None
abreuju01	Abreu	Juan	HOU	7	0	0	None
achteaj01	Achter	A. J.	LAA;MIN	45	0	0	None
acrema01	Acre	Mark	OAK	114	0	0	None
adamja01	Adam	Jason	KCA;TOR	54	0	0	None
adamsch01	Adams	Chance	NYA	16	0	0	None
adamswi02	Adams	Willie	OAK	25	0	0	None
adenhni01	Adenhart	Nick	LAA	4	0	0	None
adkinst01	Adkins	Steve	NYA	5	0	0	None

C4

question

- A person (from `people`) was a player in MLB if their `playerID` appears in `appearances`.
- A person (from `managers`) was a manager if their `playerID` appears in `managers`.
- Produce the following table from `halloffame` for people in `halloffame` that were not managers or players.
- My first 10 rows look like below.

playerid	nameLast	nameFirst	category
bulkemo99	Bulkeley	Morgan	Pioneer/Executive
johnsba99	Johnson	Ban	Pioneer/Executive
cartwal99	Cartwright	Alexander	Pioneer/Executive
chadwhe99	Chadwick	Henry	Pioneer/Executive
landike99	Landis	Kenesaw	Pioneer/Executive
connoto99	Connolly	Tommy	Umpire
klembi99	Klem	Bill	Umpire
frickfo99	Frick	Ford	Pioneer/Executive
weissge99	Weiss	George	Pioneer/Executive
gibsojo99	Gibson	Josh	Player

- Your query should produce all rows.

Answers

In [18]:

```
%%sql
SELECT
    halloffame.playerID,
    people.nameLast,
    people.nameFirst,
    halloffame.category
FROM halloffame
    LEFT JOIN
        appearances
        on halloffame.playerID = appearances.playerID
    LEFT JOIN
        managers
        on halloffame.playerID = managers.playerID
    JOIN
        people
        on halloffame.playerID = people.playerID
    WHERE
        appearances.playerID is NULL
        AND
        managers.playerID is NULL
LIMIT 10
```

```
 * mysql+pymysql://root:***@localhost
10 rows affected.
```

Out[18]:

playerID	nameLast	nameFirst	category
bulkemo99	Bulkeley	Morgan	Pioneer/Executive
johnsba99	Johnson	Ban	Pioneer/Executive
cartwal99	Cartwright	Alexander	Pioneer/Executive
chadwhe99	Chadwick	Henry	Pioneer/Executive
landike99	Landis	Kenesaw	Pioneer/Executive
connoto99	Connolly	Tommy	Umpire
klembi99	Klem	Bill	Umpire
frickfo99	Frick	Ford	Pioneer/Executive
weissge99	Weiss	George	Pioneer/Executive
gibsojo99	Gibson	Josh	Player

In [ ]:

```

```