南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

2017-2018 Academic Year
Spring Semester
Final Exam Paper

Course Name：CS307 Dept.：Computer Science Exam Duration： 120 min

Question No.	1	2	3	4	5			
Score								

This exam paper contains 5 independent questions and the score is 100 in total. (Please hand in your exam paper, answer sheet, and your scrap paper to the proctor when the exam ends.)

All printed and manuscript documents allowed, calculators allowed, other electronic devices forbidden.

Don't get stuck on a question – try to organize your time on the basis one point = one minute.

You must write your answers on the exam paper.

You are reminded of academic integrity requirements.
Papers strangely similar will get 0.

Points are indicative. They may be adjusted if one question is failed by far too many people.

Question 1: Quiz (20 points)
Question 2: Spot the mistake (18 points)
Question 3: Sizing (12 points)
Question 4: Query Analysis (25 points)
Question 5: Design (25 points)

Give at least one point for effort when the question was attempted, even if the answer is completely wrong, to make a difference with people who didn't even try.

Question 1: Quiz (20 points)

Please divide your answers on the exam paper by blocks of five, like this (w, x, y and z represent possible answers):
1-5: x,y,z,w,x
6-10: y,z,w,x,y
and so forth.

1-5 a,b,b,a,b

6-10 b,a,b,b(2points),b(2points)

11-15 b,b,d(2points),e(3points),b

Question 2: Spot the mistake (18 points)

2.1 Don't believe everything you find on the Web (8 points)

A number of websites supply "technical interview questions". The following questions and answers come from one such website:

How to find a duplicate record?

1. *duplicate records with one field*
2. *duplicate records with more than one field*

Answer.

1. *duplicate records with one field*

```
SELECT name, COUNT(email)
FROM users
GROUP BY email
HAVING COUNT(email) > 1
```

2. *duplicate records with more than one field*

```
SELECT name, email, COUNT(*)
FROM users
GROUP BY name, email
HAVING COUNT(*) > 1
```

What is wrong in the answer?

- (full grade) The GROUP BY is wrong in the first query. It should be a `GROUP BY name`
- (full grade) count(email) when grouping by email always 1
- (5 points if only this) First query would generate an error in anything

other than MySQL and SQLite

- 3 points if smelled something wrong in the first query but couldn't really say what.

2.2 Desperate Project Manager (10 points)

The following query was posted on a forum by a project manager (I have slightly modified it, but very little) and was posted with the following comment "As my developer doesn't quite master SQL and its subtleties, I'm posting this query here with the hope of getting some help. This query returns the 10 videos having the greatest number of categories in common with the video being watched (in this example video #81). This query takes 5 seconds with 2700 videos on a powerful server, we find it rather slow."

```
SELECT DISTINCT
       video_id,
       video_type,
       video_title,
       video_description,
       video_idPartner,
       video_urlMini,
       video_dateValid,
       partner_valid,
       partner_redirection,
       ( SELECT COUNT(Y.v_belongs_c_idVideo) AS NbSimilar
          FROM  v_belongs_c  Y
          WHERE Y.v_belongs_c_idVideo=81
            AND Y.v_belongs_c_idCategory IN
              (SELECT Z.v_belongs_c_idCategory
               FROM v_belongs_c Z
               WHERE Z.v_belongs_c_idVideo=video_id)) as Counter,
        ( SELECT category_singular
          FROM category,
               v_belongs_c X
          WHERE X.v_belongs_c_idVideo=video_id
            AND X.v_belongs_c_default=1
            AND category_id=X.v_belongs_c_idCategory )
                    as category_singular
FROM category,
     v_belongs_c A,
     video
     LEFT JOIN partner
            ON video_idPartner=partner_id
WHERE (A.v_belongs_c_idCategory IN
```

```
            (SELECT W.v_belongs_c_idCategory
             FROM v_belongs_c W
             WHERE W.v_belongs_c_idVideo=81)
       AND video_id=A.v_belongs_c_idVideo)
 AND (video_idPartner=0
      OR (partner_valid=1
          AND partner_redirection<>1))
 AND video_valid=1
 AND video_id <> 81
ORDER BY Counter DESC
LIMIT 10
```

WITHOUT TRYING TO REWRITE THE QUERY, can you point to a major issue in the query, and how the developer "fixed" it (hint: badly).

Problem: no join condition between category and the other tables/views in the main query (8 points).

Problem hidden by DISTINCT (2 points)

If the real problem wasn't found give points for:

- mix of different types of joins (3 points)
- Very inefficient queries partly correlated in the list of columns returned (6 points)

Question 3: Sizing (12 points)

This company wants to store in a specialized datamart information that comes from telephony systems; telephony systems are mostly today computers routing calls, which include an internal database, and generate on demand "Call Detail Records" which contain information about every phone call passed through the system (number of the caller, number called, number that answered, start time, duration, bytes of data transmitted, etc.) As most calls are passed over IP, the goal is overall to check how much bandwidth is used at different times of day, monitor the use of gateways and measure the impact of some hardware failure (would half the company be left in the dark?), predict the impact of a video-conferencing system ... and get some quantitative productivity measurements for some departments where the phone is the main tool, such as customer support and telemarketing.
Call Detail Records, or CDRs, contain on average 200 bytes. The telephony system in this (big) multinational company serves a wide geographic area and there are overall around 200,000 calls a day. You are reminded that data blocks contain a header and various overhead, and that raw data will use around 15%

more storage when inserted into a database table. Additionally, an estimated 60% of table storage will be used by indexes. This will be the "big table" in the database, the remainder (data dictionary, other tables, system storage) can be estimated at around 400M.
This system is a decision support system and will not be used 24x7. Additionally, data is obtained from the telephony system computers (Cisco call them "Call Managers") that can generate on demand CDR files for the past 10 days. As a consequence, a daily cold backup every night is considered to be quite sufficient, as it takes about 20 minutes to load and process daily data from a CDR file. However, many batch processes are running during the night and the maintenance window is narrow.
We have relatively slow disks that allow transferring about 60M/s. We want to be able to backup or restore the database in 30 minutes or less. How long can we keep data online before archiving it? In other words, how far back in time will we be able to go if we want to keep the size such as we can backup or restore the database in under 30 minutes?

Each record = 200 bytes x 200,000 records a day = 40M/day of raw data
Add 15 % overhead = 6M/day => 46M/day (2 points for this)
Indexes 60% => 28M/day (rounded up)
So we are adding 74M/day to the database. (+3 points)

If we can backup 60M/s we can backup 60 * 60 = 3600M per minute
or 96000M in 30 minutes (+ 2 points). If we remove the 400M of other data, that's about 95,500M of CDR data that we can backup in 30 minutes, about 1,300 days or about 3 and a half years (+ 3 points - give full points for anything in the right order of magnitude, it's just a rough assessment)

Question 4: Query analysis (25 points)

The following figure describes a database used to store information about the states and union territories of India, as well as the official languages in use.

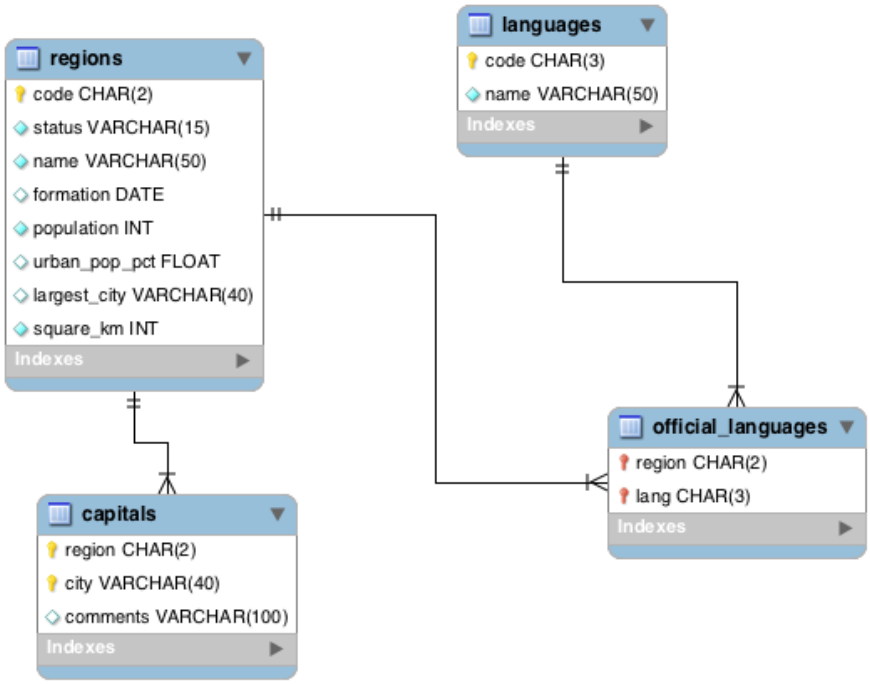

Table Descriptions

regions

code	Two-letter code	Official code (in uppercase) for the state or the region. Always populated.
status	String	Tells whether the region is a state or a union territory, contains either *State* or *Union Territory*. Always populated.
name	String	Name of the region. Always populated.
formation	Date	Date of formation for states. Empty for union territories.
population	Integer	Population of the region. Always populated.
urban_pop_pct	Integer	Percentage of the population of the region that lives in cities.
largest_city	String	Name of the largest city in the region. Empty if the largest city is the capital.
square_km	Integer	Area of the region, expressed in square kilometers. Always populated.

capitals

region	Two-letter code	Official code in uppercase for the region, matches the code column in the region table. Always populated.
city	String	Name of the capital. Always populated.
comments	String	Special comment for unusual cases, such as capitals shared by two regions. Usually empty.

official_languages

region	Two-letter code	Official code in uppercase for the region, matches the code column in the region table. Always populated.
lang	Three-letter code	Lowercase code of a language, matches the code column in the languages table. Always populated.

languages

code	Three-letter code	Standard code in lowercase for a language. Always populated.
name	String	Name of the language. Always populated.

For the following questions, give the identifier of the queries that correctly answer the questions, "None" if no query gives the correct answer. The queries were written with SQLite and are assumed to run.

1 point for everybody who attempted the questions.
2 points per correctly picked query, 2 points per correctly NOT picked query (if the right answer is query 1 and the student chooses query 3, 2 points for not picking query 2 that is wrong)

A	What is the total population of states and territories where Hindi is an official language?

A.1	`SELECT SUM(r.population)`
	`FROM regions r`
	`  INNER JOIN official_languages ol`
	`  ON ol.region = r.code`
	`  INNER JOIN languages l`
	`  ON l.code = ol.lang`
	`WHERE l.name = 'Hindi'`

A.2	`select la.name, reg.population`
	`from (select re.code, re.population`
	`  from regions re`
	`  group by re.code) reg`
	`  join official_languages of`
	`  on of.region = reg.code`
	`  join languages la`
	`  on la.code = of.lang`
	`where la.name = 'Hindi'`

A.3	`select sum(population)`
	`from regions r`
	`  join official_languages ol`
	`  on r.code = ol.region`
	`  join languages l`
	`  on ol.lang = l.code`
	`where l.name = 'Hindi'`

Correct answers: A1 and A3

B	What are the capitals that aren't the only capital of a single state or territory (some capitals are shared, some states have several capitals)

	- the query should return TWO columns 1) comma-separated list of capitals (when there are several capitals 2) Comma-separated list of states (when several states)

B.1	`SELECT group_concat(c.city),group_concat(c.region)`
	`FROM capitals c`
	`GROUP BY c.city, c.region`
	`HAVING count(c.city) > 1`
	`OR count(c.region) > 1`

B.2	`select group_concat(ca.city) cities,`
	`group_concat(reg.name) regions`
	`from (select re.code, re.name`
	`from regions re`
	`group by re.code) reg`
	`join capitals ca`
	`on ca.region = reg.code`
	`where ca.comments != ''`

B.3	`select r.name, c.city, count(*) c_count`
	`from regions r`
	`join capitals c`
	`on r.code = c.region`
	`group by c.city`
	`having c_count > 1`
	`union`
	`select r.name, c.city, count(*) s_count`
	`from regions r`
	`join capitals c`
	`on r.code = c.region`
	`group by r.name`
	`having s_count > 1`

Correct answers: None

C	What are the name, population, and number of languages of the state(s) or region(s) with the most official languages?

C.1	`SELECT r.name, SUM(population), COUNT(ol.lang) AS langCount`
	`FROM regions r`
	`INNER JOIN official_languages ol`
	`ON ol.region = r.code`
	`WHERE langCount = (SELECT COUNT(ol.lang) AS langCount2`
	`FROM official_languages ol`
	`GROUP BY ol.region`
	`ORDER BY langCount2 desc`
	`LIMIT 1)`
	`GROUP BY r.name`
	`ORDER BY langCount desc`

C.2	`select re.name, re.population, off.numOfLang`
	`from (select of.region, count(lang) numOfLang`
	`from official_languages of`
	`group by of.region) off`
	`join regions re`
	`on re.code = off.region`

C.3	`select r.name, r.population, count(*) langs`
	`from regions r`
	`join official_languages ol`
	`on r.code = ol.region`
	`left join languages l`
	`on ol.lang = l.code`
	`group by r.code`
	`having langs = (select max(n_langs)`
	`from (select count(*) n_langs`
	`from regions r`

	join official_languages ol on r.code = ol.region group by r.code))

Correct answer: C3

D	What are the States (<u>not</u> Union Territories) where English is NOT an official language? Display their names in alphabetical order.
D.1	SELECT DISTINCT r.name FROM regions r INNER JOIN official_languages ol ON ol.region = r.code INNER JOIN languages l ON l.code = ol.lang WHERE r.status = 'State' AND l.name <> 'English' ORDER BY r.name
D.2	select re.name from (select of.region, of.lang from official_languages of group by of.region) off join regions re on re.code = off.region join languages la on la.code = off.lang where re.status = 'State' and la.name != 'English'

(D.3 follows on next page)

D.3	`select r.name, l.name`
	`from regions r`
	`  join official_languages ol`
	`  on r.code = ol.region`
	`  join languages l`
	`  on ol.lang = l.code`
	`group by r.name`
	`having l.name != 'English'`
	`  and r.status = 'State'`
	`order by r.name asc`

Correct answer: None

Question 5: Design (25 points)

We want to create a music database. Usually you have a relationship between music album and artists. We'll just focus here on artists. The problem of artists is that an album can be credited to an individual artist (say " Jacky Cheung (张学友)" or a band (say "Phoenix Legend (凤凰传奇)" or "Beyond"); so a band can also be considered an artist. We may want for individuals to record date of birth and (possibly) of death, and for bands the date of creation and (possibly) the date when the band was officially disbanded. Additionally, we may want to record who are the members of a pop band (perhaps not of a symphonic orchestra), and band members may change over time. Band members can also move to other bands, or start a solo career.

5.1. How would you model what is described above? (20 points)

12 points for artists + persons + bands, minus 2 per constraint (PK/FK) not specified

- "Parent" table artists artistid, type (band/person), optionally name (or surname/given_name), optionally birthdate, deathdate ("death" been disbanding for bands)
- Table persons
  - personid PK and also FK references artists(artistid)
  - surname/given name might be there
  - dates might also be there
- Table bands
  - bandid PK and also FK references artists(artistid)
  - name might be there
  - dates might also be there

+ 8 points for band membership, -1 per missing constraint, -2 if pk is only (personid, bandid).

- Band_membership
    - personid foreign key references persons(personid)
    - bandid foreign key references bands(bandid)
    - member_from date not null
    - member_until date (can be null)
    - primary key (personid, bandid, member_from)

  Somebody might leave a band and return.

12 points if single artists table with type, and band_membership table with two distinct foreign keys referencing the artistid column. +3 if idea of a trigger for checking types on insert (not a good idea, but good to think of checking consistency)

5.2. Would you make the artist identifier an attribute of an album, or would you use a relationship table (artistid, albumid)? Justify.

Shoud use a relationship table (+1). Justification is that several artists can be associated with a single album (duets, xxx and friends ...) (+4)