香 港 中 文 大 學
The Chinese University of Hong Kong

Midterm Examinations Second Term, 2021-2022

Course Code & Title : SEEM3550A Fundamentals in Information Systems

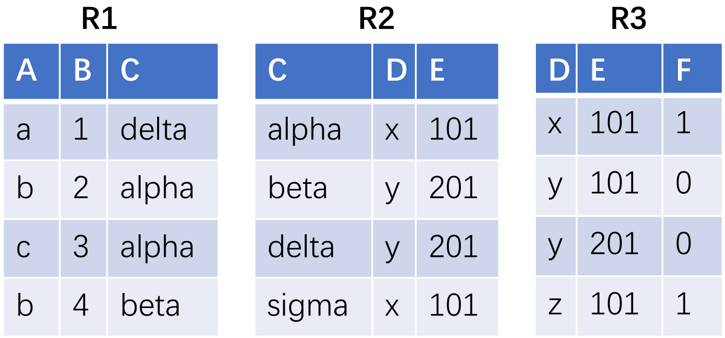
Time allowed : 2 hours 0 minutes

Student I.D. No. : .... **Seat No.** : ....

- **Q1. Keys**[20 marks]

  Consider the relational databases shown as below:

  - $R1(A, B, C)$
  - $R2(C, D, E)$
  - $R3(D, E, F)$



R1

A	B	C
a	1	delta
b	2	alpha
c	3	alpha
b	4	beta

R2

C	D	E
alpha	x	101
beta	y	201
delta	y	201
sigma	x	101

R3

D	E	F
x	101	1
y	101	0
y	201	0
z	101	1

  - (a) [10 marks] **Superkey**: Let $K \subseteq R$, K is a superkey of R if values for K are sufficient to identify a unique tuple of each possible relation $r(R)$. List all the superkeys of each above relation.

    Ans: $R1 : \{B\}, \{A, C\}, \{A, B\}, \{A, B, C\}, \{B, C\}$
    $R2 : \{C\}, \{C, D\}, \{C, E\}, \{C, D, E\}$
    $R3 : \{D, E\}, \{D, E, F\}$

- (b) [3 marks] **Candidate key**: Superkey K is a candidate key if K is minimal. List all the candidate keys of each above relation.

  Ans: $R1 : \{B\}, \{A, C\}$
  $R2 : \{C\}$
  $R3 : \{D, E\}$

- (c) [3 marks] **Primary key**: One of the candidate keys is selected to be the primary key. List the primary keys of each above relation.

  Ans: $R1 : \{B\} or \{A, C\}$
  $R2 : \{C\}$
  $R3 : \{D, E\}$

- (d) [4 marks] List the results when you do a natural join between R2 and R3.

  Ans: {alpha, x, 101, 1}
  {beta, y, 201, 0}
  {delta, y, 201, 0}
  {sigma, x, 101, 1}

- **Q2. Relational Algebra [20 marks]**

  Consider the following relations about NBA:

  - $PLAYER(\underline{pid}, pname, nation)$. Each tuple describes a player with *pid* being the player's id, *pname* his name, and *nation* his nationality. The table has a primary key *pid*.
  - $TEAM(\underline{tid}, tname)$. Each tuple describes a team with *tid* being the team's id, and *tname* its name. The table has a primary key *tid*.
  - $REGISTER(\underline{pid}, tid, salary, \underline{year})$. Each tuple records the fact that a certain player (indicated by *pid*) played for a certain team (indicated by *tid*) in a specific *year* with an annual income given in *salary*. The table has a primary key (*pid,year*). Note that a player may belong to various teams in different years.

  Represent the following queries with relational algebra. Some useful relational operators are: σ for selection, Π for projection, $\bowtie$ for natural join, $\times$ for cartesian product, $-$ for set difference, G for aggregation.

  - (a) [5 marks] Find the ids of players who have played for the team with name "ABC".

    Ans: $\Pi_{pid}(\sigma_{tname='ABC'}(TEAM \bowtie REGISTER))$

  - (b) [5 marks] Find the ids of players who have never played for the team with name "ABC". (Hints: You could (not a must) use set difference "-")

    Ans: $\Pi_{pid}(PLAYER) - \Pi_{pid}(\sigma_{tname='ABC'}(TEAM) \bowtie REGISTER))$

- (c) [5 marks] For each team, find the team id and the average salary of its players. (Hints: You could (not a must) use aggregate operation G and aggregation function $avg()$)

  Ans: ${}_{tid}G_{avg(salary)}(REGISTER)$

- (d) [5 marks] Find the names of the teams that player named "DEF" ever played for.

  Ans: $\Pi_{tname}(TEAM \bowtie REGISTER \bowtie \Pi_{pid}(\sigma_{pname='DEF'}(PLAYER)))$

- **Q3 [25 marks] SQL (Basics)**

Consider the relational databases shown as below, where the primary keys are underlined:

- Student(**ID**, name, dept_name)
- Exam(**ID**, score)

The student table contains all students' basic information including name and department name. The primary key is student ID.

The exam table stores each student's score in an exam. Note that ***some students do not take this exam***, hence these students' IDs are not in the exam table. The primary key is student ID.

Write SQL statements to solve the following queries:

- (a) [5 marks] Show the ID of the students who doesn't pass the exam ($score < 60$).

  Select ID
  From Exam
  Where $score < 60$;

- (b) [5 marks] Show the ID, name and score for each student who takes the exam and his name contains "Bob".

  Select ID, name, score
  From Student, Exam
  Where Student.ID = Exam.ID and name like $'$%Bob%$'$;

- (c) [5 marks] Show the ID, name and score for each student who passes the exam. The results should be listed in descending order of score. (pass: $score \geq 60$)

  Select ID, name, score
  From Student, Exam
  Where Student.ID = Exam.ID and $score \geq 60$
  Order by score DESC;

- (d) [5 marks] Show the ID for each student who is in the math department **or** passes this exam.

In the math department: dept_name = “math”
Pass the exam: $score \geq 60$

Select ID from Student where dept_name = $'$math$'$
UNION Select ID from Exam where $score \geq 60$;

- (e) [5 marks] Show the department name and the number of students taking this exam if the number is greater than ten.

  Select dept_name, count(*)
  From Student, Exam
  Where Student.ID = Exam.ID
  Group by dept_name
  Having $count(*) > 10$;

- **Q4. [35 marks] SQL (Group By and Nested Subqueries)**

  You are given the following three relations in a university database as below (with their primary keys underlined):

  - $INSTRUCTOR(\underline{Iid}, Name, DeptName)$.
  - $ADVISOR(\underline{Sid}, Iid)$.
  - $STUDENT(\underline{Sid}, Name, DeptName)$.

  The relation ADVISOR records the advisor of each a student. You may assume that every student has an advisor. Iid and Sid are of type `int`, and other attributes are of type `varchar(30)`.

  Write SQL statements to solve the following queries:

  - (a) [5 marks] Show the total number of students in Student table.

    Select count(*)
    From STUDENT;

  - (b) [5 marks] Show the DeptName and the number of instructors in this department.

    Select DeptName, count(*)
    From Instructor
    Group by DeptName;

  - (c) [5 marks] Show the average number of instructors in a department.

```
Select avg(INum)
From ( Select count(*) as INum
         From INSTRUCTOR
         Group by DeptName);
```

- (d) [5 marks] For each department, show the DeptName and the number of instructors who advise the students in that department.

```
Select DeptName, count(*)
From ADVISOR natural join STUDENT
Group by DeptName;
```

- (e) [5 marks] Show the id of instructors who advise more than 1 student.

```
Select Iid
From ADVISOR AS A
group by Iid
having count(*)>1
```

- (f) [5 marks] Show the names of the instructors who advise more than 1 student. If multiple instructors have the same names, output the name only for one time.

```
Select DISTINCT Name
From (Select Iid
From ADVISOR AS A
group by Iid
having count(*)>1) natural join INSTRUCTOR

Select DISINCT, Name
From ADVISOR natural join INSTRUCTOR
group by Iid, Name
having count(*)>1)

Select DISTINCT Name
From INSTRUCTOR
Where Iid in (Select Iid
From ADVISOR AS A, STUDENT AS S
Where S.Sid = A.Sid
group by Iid
having count(*)>1)

Select DISTINCT Name
From INSTRUCTOR as I
Where Exist (Select Iid
From ADVISOR AS A, STUDENT AS S
Where S.Sid = A.Sid and I.Iid=A.Iid
group by Iid
having count(*)>1)

Select DISTINCT N.Name
From ADVISOR AS A, ADVISOR AS B, INSTRUCTOR as N
Where A.Iid = B.Iid and A.Sid > B.Sid and N.Iid = A.Iid
```

- (g) [5 marks] Show the Iids of the instructors who advise no students.

```
Select Iid
From INSTRUCTOR
Where Iid not in (Select Iid From ADVISOR)
```