IDMP_ASSSIGNMENT_2

Name: Ayush Patel

NUID: 00275119

Q1.

1)

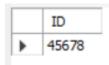
Code:

```
#Q1. 1)#
SELECT dept_name, max(salary) as max_sal
    FROM instructor
GROUP BY dept_name
ORDER BY max_sal DESC;
```

	dept_name	max_sal
١	Physics	95000.00
	Comp. Sci.	92000.00
	Finance	90000.00
	Elec. Eng.	80000.00
	Biology	72000.00
	History	62000.00
	Music	40000.00

Code:

Output:



3)

Code:

```
#Q1. 3)#
SELECT DISTINCT section.course_id , course.title
FROM course, time_slot, section
WHERE course.dept_name = 'Comp. Sci.'
   AND time_slot.end_hr >= 12
   AND course.course_id = section.course_id
   AND time_slot.time_slot_id = section.time_slot_id;
```

	course_id	title
•	CS-101	Intro. to Computer Science
	CS-315	Robotics

Code:

```
#Q1. 4)#
SELECT p.title as prereq_title,p.course_id AS prereq_id
FROM course as c,course as p, prereq
WHERE c.course_id = prereq.course_id
    AND p.course_id = prereq.prereq_id
    AND c.title = 'Robotics';
```

Output:

	prereq_title	prereq_id
•	Intro. to Computer Science	CS-101

5)

Code:

```
#Q1. 5)#
SELECT id, name
FROM instructor
WHERE salary=(SELECT MAX(salary) FROM instructor);
```

	id	name
•	22222	Einstein

Code:

```
#Q1. 6)#
SELECT course_id, sec_id, COUNT(ID) AS student_nums
FROM section NATURAL JOIN takes
WHERE semester = 'Spring'
    AND year = 2017
GROUP BY course_id, sec_id;
```

Output:

	course_id	sec_id	student_nums
•	CS-190	2	2
	EE-181	1	1

7)

Code:

```
#Q1. 7)#

SELECT section.course_id, section.sec_id, COUNT(ID) AS student_nums
FROM section LEFT JOIN takes ON section.course_id = takes.course_id

AND section.sec_id = takes.sec_id

AND section.semester = takes.semester

AND section.year = takes.year

WHERE section.semester = 'Spring'

AND section.year = 2017

GROUP BY section.course_id, section.sec_id;
```

course_id	sec_id	student_nums
CS-190	1	0
CS-190	2	2
EE-181	1	1

Code:

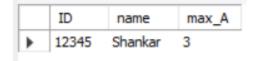
```
#Q1. 8)#
SELECT instructor.ID, instructor.name, COUNT(course_id) AS no_of_courses
FROM instructor, teaches
WHERE teaches.ID = instructor.ID
GROUP BY name, ID
HAVING COUNT(course_id) >= 3;
```

Output:

	ID	name	no_of_courses
•	10101	Srinivasan	3
	83821	Brandt	3

9)

Code:



Code:

Output:

```
ID name

▶ 19991 Brandt
```

11)

Code:

```
#Q1. 11)#

SELECT instructor.ID, instructor.name

FROM instructor

WHERE instructor.ID NOT IN (

(SELECT DISTINCT teaches.ID

FROM teaches LEFT JOIN takes ON (teaches.course_id = takes.course_id

AND teaches.semester = takes.semester

AND teaches.sec_id = takes.sec_id

AND teaches.year = takes.year)

WHERE takes.grade = 'A'));
```

	ID	name
•	12121	Wu
	15151	Mozart
	22222	Einstein
	32343	El Said
	33456	Gold
	45565	Katz
	58583	Califieri
	76543	Singh
	98345	Kim

Code:

```
#Q1. 12)#
SELECT instructor.ID, instructor.name
FROM instructor
WHERE instructor.ID NOT IN
    (SELECT DISTINCT teaches.ID
        FROM teaches LEFT JOIN takes ON (teaches.course_id = takes.course_id
                                AND teaches.semester = takes.semester
                                AND teaches.sec_id = takes.sec_id
                                AND teaches.year = takes.year)
        WHERE takes.grade = 'A' )
AND instructor.ID IN
    (SELECT DISTINCT teaches.ID
        FROM teaches LEFT JOIN takes ON (teaches.course_id = takes.course_id
                                AND teaches.semester = takes.semester
                                AND teaches.sec_id = takes.sec_id
                                AND teaches.year = takes.year)
        WHERE takes.grade IS NOT NULL);
```

	ID	name
•	32343	El Said
	12121	Wu
	22222	Einstein
	45565	Katz
	15151	Mozart
	98345	Kim

Code:

```
SELECT student.ID, student.name, course_id
FROM student LEFT JOIN takes ON student.ID = takes.ID
GROUP BY student.id, takes.course_id
HAVING COUNT(takes.course_id) >= 2;
```

Output:



14)

Code:

```
#Q1. 14)#
SELECT student.ID, student.name, course_id
FROM student LEFT JOIN takes ON student.ID = takes.ID
GROUP BY student.id, takes.course_id
HAVING COUNT(DISTINCT takes.course_id) >= 3;
```



Code:

```
#Q1. 15)#
SELECT instructor.ID, instructor.name, instructor.dept_name
FROM instructor JOIN teaches ON instructor.id = teaches.id
GROUP BY instructor.id, instructor.dept_name
HAVING COUNT(*) =
( SELECT COUNT(*)
    FROM course
WHERE course.dept_name = instructor.dept_name);
```

Output:

	ID	name	dept_name
•	98345	Kim	Elec. Eng.
	12121	Wu	Finance
	32343	El Said	History
	15151	Mozart	Music
	22222	Einstein	Physics

Q2.

1)

Code:

```
#Q2. 1)#
INSERT INTO course
VALUES('CS-001','Weekly Seminar','Comp. Sci.',2);
```

```
Action Output

    210 14:59:10 insert into prereq values ('BIO-399', 'BIO-101')

                                                                                                                       1 row(s) affected
                                                                                                                                                                                                                        0.015 sec

    211 14:59:10 insert into prereq values ('CS-190', 'CS-101')

                                                                                                                       1 row(s) affected
                                                                                                                                                                                                                        0.000 sec
212 14:59:10 insert into prereq values (CS-315', 'CS-101')
                                                                                                                       1 row(s) affected
                                                                                                                                                                                                                        0.016 sec

    213 14:59:10 insert into prereq values ('CS-319', 'CS-101')

                                                                                                                                                                                                                        0.000 sec
                                                                                                                       1 row(s) affected
214 14:59:10 insert into prereq values (CS-347, 'CS-101')
                                                                                                                                                                                                                        0.015 sec
                                                                                                                       1 row(s) affected

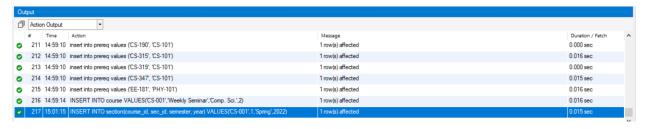
    215 14:59:10 insert into prereq values ('EE-181', 'PHY-101')

                                                                                                                       1 row(s) affected
                                                                                                                                                                                                                        0.016 sec
```

Code:

```
#Q2. 2)#
INSERT INTO section(course_id, sec_id, semester, year)
VALUES('CS-001',1,'Spring',2022);
```

Output:



3)

Code:

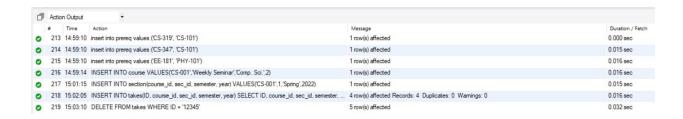
```
#Q2. 3)#
INSERT INTO takes(ID, course_id, sec_id, semester, year)
SELECT ID, course_id, sec_id, semester, year
FROM student, section
WHERE dept_name = 'Comp. Sci.'
    AND course_id = 'CS-001';
```



Code:

```
#Q2. 4)#
DELETE FROM takes
WHERE ID = '12345';
```

Output:

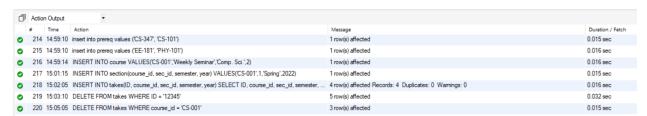


5)

Code:

```
#Q2. 5)#
DELETE FROM takes
WHERE course_id = 'CS-001';
```

Output:



Observation:

CS-001 has been deleted from section and enrollment as well.

Q3.

Code:

```
#Q3#
use insurance;

create table person(
    driver_id varchar(8) primary key not null,
    name varchar(20),
    address text);
```

Output:

Table: person

Columns:

driver_id varchar(8) PK varchar(20) address text

Assumptions:

- I have taken driver_id as varchar(8), according to the US driver license it has alphabetic as well as numeric values.
- Name is also varchar(20) as it may include First and last name.
- Address as text because it may have alphabetic, numeric and special characters such as (.,()&/,etc.)

Code:

```
create table car(
    license_plate varchar(6) primary key not null,
    model varchar(10),
    year numeric(4,0));
```

Output:

```
Columns:
license_plate warchar(6) PK warchar(10) year decimal(4,0)
```

Assumptions:

- License_plate is varchar(6) as it can have alpha-numeric values for eg.(BOS 123).
- Model can also have the same alpha-numeric values such as Tesla Model S,3,X,Y.
- Year is numeric here with no decimal for eg. (2000, 2022, etc.)

Code:

```
create table owns(
    driver_id varchar(8),
    license_plate varchar(6),
    primary key(driver_id, license_plate),
    foreign key(driver_id) references person(driver_id),
    foreign key(license_plate) references car(license_plate));
```

Output:

Table: owns

Columns:

driver_id varchar(8) PK license plate varchar(6) PK

Assumptions:

Here, the driver_id and license_plate are the foreign keys for person and car in which they are the primary keys. They are also the primary key here.

Code:

```
create table accident(
    report_number int primary key auto_increment,
    date date,
    location varchar(20));
```

Output:

Table: accident

Columns:

report_number date int AI PK date location varchar(20)

Assumptions:

- Report_number is integer value and is set as auto increment.
- Date has a 'date' format in SQL.
- Location is varchar(20) for eg. (Los Angeles, Boston, etc.)

Code:

```
create table participated(
   damage_amount numeric(10,2),
   report_number int,
   driver_id varchar(8),
   license_plate varchar(6),
   primary key(report_number, license_plate),
   foreign key (report_number) references accident(report_number),
   foreign key (license_plate) references car(license_plate),
   foreign key (driver_id) references person(driver_id));
```

Output:

```
Columns:

damage_amount decimal(10,2)
report_number int PK
driver_id varchar(8)
license_plate varchar(6) PK
```

Assumptions:

- The damage_amount is set for 10 digits with upto 2 decimal places.
- Here, the report_number and license_plate are the foreign keys for accident and car in which they are the primary keys. They are also the primary key here.
- Driver_id is the foreign key for this table which is the primary key for person.