Answer Sheet by Pejman: Test 1

A:

A schema describing theatres, cities where they are located and shows is defined as follows: CITY (Name, State, Country) THEATRE (Name, City, State, Capacity) SHOW (Title, Artist, Hall, Attendance) Write the following queries in SQL

i) Find names of artists who performed before at least 5000 people, together with cities where those performances took place.

```
SELECT DISTINCT Artist, City

FROM SHOW

INNER JOIN THEATRE ON SHOW.Hall = THEATRE.Name AND SHOW.City = THEATRE.City

INNER JOIN CITY ON THEATRE.City = CITY.Name

WHERE Attendance >= 5000;
```

ii) Find all states in Japan where Mr. X has performed.

```
SELECT DISTINCT State

FROM SHOW

INNER JOIN THEATRE ON SHOW.Hall = THEATRE.Name AND SHOW.City = THEATRE.City

INNER JOIN CITY ON THEATRE.City = CITY.Name

WHERE Artist = 'Mr. X' AND Country = 'Japan';
```

iii) List all artists who never played in Delhi.

```
SELECT DISTINCT Artist

FROM SHOW

WHERE City != 'Delhi';
```

iv) Find the name of theatres in Tokyo whose capacity exceeds 5000

```
SELECT DISTINCT Name
FROM THEATRE
INNER JOIN CITY ON THEATRE.City = CITY.Name
WHERE Capacity > 5000 AND CITY.Country = 'Japan' AND CITY.State = 'Tokyo';
```

B

Consider the insurance database given below: PERSON (driver-id, name, address) CAR (license, model, year) ACCIDENT (report-number, date, location) OWNS (driver-id, license) PARTICIPATION (driver-id, report-number, damage-amount) Construct the following SQL queries for this relational database.

i) Find the total number of people who owned cars that were involved in accidents in 2004.

```
SELECT COUNT(DISTINCT driver_id)
FROM OWNS
INNER JOIN CAR ON OWNS.license = CAR.license
INNER JOIN ACCIDENT ON CAR.license = ACCIDENT.license
WHERE YEAR(ACCIDENT.date) = 2004;
```

ii) Find the number of accidents in which the cars belonging to “Vincent” were involved.

```
SELECT COUNT(DISTINCT report_number)
FROM OWNS
INNER JOIN CAR ON OWNS.license = CAR.license
INNER JOIN ACCIDENT ON CAR.license = ACCIDENT.license
WHERE OWNS.driver_id = (
  SELECT driver_id FROM PERSON WHERE name = 'Vincent'
);
```

iii) Delete the Toyota belonging to “John”.

```
DELETE FROM CAR
WHERE license IN (
  SELECT license FROM OWNS WHERE driver_id = (
    SELECT driver_id FROM PERSON WHERE name = 'John'
  )
) AND model = 'Toyota';
```

iv) Find the top 5 damages amount paid out last year. Show Driver info, report number and license appropriately.

```
SELECT P.name, A.report_number, O.license, participation.damage_amount
FROM PERSON P
INNER JOIN OWNS O ON P.driver_id = O.driver_id
INNER JOIN PARTICIPATION ON P.driver_id = PARTICIPATION.driver_id
INNER JOIN ACCIDENT A ON PARTICIPATION.report_number = A.report_number
WHERE YEAR(A.date) = YEAR(CURRENT_DATE - INTERVAL 1 YEAR)
ORDER BY PARTICIPATION.damage_amount DESC
LIMIT 5;
```

C

Consider the following tables which give details of customers, trucks and packets booked by customers, which are carried by trucks and write SQL commands to do the following tables: CUSTOMER (c_no, c_name, c_address) TRUCK (t_no, driver_name) PACKET (p_no, c_no, t_no, date_of_booking, weight, destination) Write these queries:

i) Destinations which have received more than 10 packets.

```
SELECT destination
FROM PACKET
GROUP BY destination
HAVING COUNT(*) > 10;
```

ii) Name of Customers who have sent at least one packet of weight more than one kg to 'SINGAPORE'.

```
SELECT DISTINCT c_name
FROM CUSTOMER
INNER JOIN PACKET ON CUSTOMER.c_no = PACKET.c_no
WHERE PACKET.destination = 'SINGAPORE' AND PACKET.weight > 1;
```

iii) Name of all Customers whose packets were delivered by a driver whose name is 'Mark'. iv) Three top customers (names) in terms of total packet weight sent by them (list is to be in descending order of total weight).

```
SELECT DISTINCT c_name
FROM CUSTOMER
INNER JOIN PACKET ON CUSTOMER.c_no = PACKET.c_no
INNER JOIN TRUCK ON PACKET.t_no = TRUCK.t_no
WHERE TRUCK.driver_name = 'Mark';
```

v) Name of all Customers whose individual shipments are less than one kg.

```
SELECT c_name, SUM(weight) AS total_weight
FROM CUSTOMER
INNER JOIN PACKET ON CUSTOMER.c_no = PACKET.c_no
GROUP BY c_name
ORDER BY total_weight DESC
LIMIT 3;
```