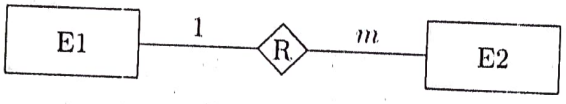
Master of Computer Applications
MCAE403: Database Applications
Unique Paper Code: 223402403
Semester IV
May-June-2022
Year of admission: 2020

Time: 3.5 Hours **Max. Marks: 70**

Note:
1. Attempt all parts of a question together.
2. Do only 7 out of 8 questions.

Q.1 (a) Consider the following entity relationship diagram (ERD), where two entities E1 and E2 have a relation R of cardinality 1:m. (4)



The attributes of E1 are A11, A12, and A13 where A11 is the key attribute. The attribute of E2 are A21, A22 and A23 where A21 is the key attribute and A23 is a mufti-valued attribute. Relation R does not have any attribute. A relational database containing minimum number of tables with each table satisfying the requirements of the third normal form (3NF) is designed from the above ERD. Write the number of tables required to represent above ER diagram. Also, write the names of the tables with their attributes.

(b) Differentiate between
(i) ALL and ANY logical operators. (4)
(ii) row-type variable and record types.
Explain by giving proper examples.

(c) Write a PL/pgSQL code to get the actor count in a variable *actor_count* by using the table actor. (2)
Table Structure:
actor (actor_id, first_name, last_name, last_update)

Q.2 The functional dependencies for relation $R = \{A, B, C, D, E, F, G, H, I, J\}$ are
$F = \{AB \rightarrow C,\ BD \rightarrow EF,\ AD \rightarrow GH,\ A \rightarrow I,\ H \rightarrow J\}$
(a) List all candidate keys of the above relation.
(b) Find the highest normal form the R is in. (2.5)
(c) Convert the relation into the highest normal form (BCNF). (2.5)
(d) Check whether the decomposition is loss-less and functional dependency preserving. (2.5) (2.5)

Q.3: Refer the relation given below:

City(City_id, city, country_id, last_update)

(a) Find the difference between the total number of city entries in the table and the number of distinct city entries in the table. (2)

(b) Find the duplicate city entries in the city table. Further, write another query to delete duplicate city entries. (3)

(c) Query the two cities in city table with the shortest and longest CITY names, as well as their respective lengths (i.e.: number of characters in the name). If there is more than one smallest or largest city, choose the one that comes first when ordered alphabetically. (3)

(d) Write a query to get the cities name in the upper case using the alias name as '*mycity*'. Also, write a query to fetch *today's* date. (2)

Q.4: (a) The following table has two attributes *Attribute1* and *Attribute2.* Here, *Attribute2* is the foreign key referencing attribute with on-delete cascade. (2)

Attribute1	Attribute2
A	C
B	C
C	B
D	A
F	A
G	D
E	C

List out all the tuples that must be additionally deleted to preserve referential integrity when the *tuple (A, C)* is deleted.

(b) Find the total number of customers per country and city. Further, Write another query to show list of all the different values in the *Customers* table for *contact_title.* Include count for each *contact_title.* (2)

Table Structure:

Customers(customer_id, company_name, contact_name, contact_title, address:, city, region, postal_code, country,phone, fax)

(c) Show the date of the first order ever made in the *Orders* table. (1)

Table Structure:

Orders(order_id, customer_id, order_date, shipped_date, ship_via, ship_name)

(d) Write a query to show first name, last name, birth date and hire date of only those employees that have the title of Sales Representative and also are in the United States. Order the results by birth date, so we have the oldest employee first. In the output of the above query, show only the date portion of the birth date field. (2)

Table Structure:

Employees(employee_id, last_name, first_name, title, birthdate, hiredate, city, country, postal_code)

e) Create a new unlogged table 'temp_info' from the result of *4(d).* (3)

Q.5: (a) Explain Trigger and Trigger types. (4)

(b) Suppose there are two tables *employee_info* and *employee_audits.* employee_info contains information about employees of a company and has fields *(id, first_name, last_name). employee_audits* has fields (*id, employee_id, last_name, changed_on*) and updated when the last name of an employee changes. (6)
(i) Create a trigger function *log_last_name_changes* to log the changes in *employee_audits* table when the last name of an employee changes.
(ii) Bind the trigger function to the *employee_info* table.

Q.6: (a) How Stored procedures are different from functions? Write at-least four differences. (4)

(b) Write a PL/pgSQL function to return all films whose titles match a particular pattern. For example: if the pattern is "*Th*", the function should return all the movies whose names start with '*Th*'(case-insensitive). (6)
Further, write a command to drop a stored procedure *insert_actor* and its related objects from the database. If *insert_actor* stored procedure is not in the database, then the command should not throw an error.
Table Structure:
Film(film_id, title, release_year, rating)

Q.7: (a) What is Coalesc function? Write its usage. Further, write the output of the following PostgreSQL query using *Items* table.: (3)

```
SELECT product, (price - COALESCE(discount,0)) AS net_price FROM items
```

Items table:

	id	product	price	discount
0	1	A	1000.0	10.0
1	2	B	1500.0	20.0
2	3	C	800.0	5.0
3	4	D	500.0	NaN

(b) Write a query to compare the average order amount for each store with the minimum and maximum of the average order amount of all stores. (4)
Table structure:
orders_store(Id, date, customer_id, store, employee_id, amount)
Further, Your output should look like below:

	store	average_order	min_avg_order	max_avg_order
0	Center	489.666667	338.5	725.0
1	East	338.500000	338.5	725.0
2	West	725.000000	338.5	725.0

(c) Write a query to display the Rating expression as per following convention : (3)

G: General Audience, **PG:** Parental Guidance Suggested,
PG-13: Parents Strongly Cautioned, **R:** Restricted, **NC-17:** Adults Only.

Table Structure:
Film(film_id,title, release_year, rating)

Q.8: (a) Consider the following relation schema and set of functional dependencies:
Emp_Dept (SIN, E_Name, B_Date, Address, D_Num, D_Name, D_Manager)
F = { *SIN → {E_Name, B_Date, Address, D_Num}, D_Num → {D_Name, D_Manager}*}

(i) Calculate $\{SIN\}^+$ and $\{D\_Num\}^+$ with respect to F. (1)

(ii) Is the set of functional dependencies F minimal? If not, try to find a minimal set of functional dependencies that is equivalent to F (minimal cover). (2)

(iii) What anomalies can happen to *Emp_Dept* relation? Give examples. (3)

(b) What are Grouping SET, ROLLUP and CUBE in PostgreSQL? Further, write the output of the following query using *Sales* Relation: (4)

```
SELECT brand, segment, SUM (quantity)
FROM sales
GROUP BY CUBE (brand, segment)
ORDER BY brand, segment;
```

Sales Table:

	brand	segment	quantity
0	ABC	Premium	100
1	ABC	Basic	200
2	XYZ	Premium	100
3	XYZ	Basic	300

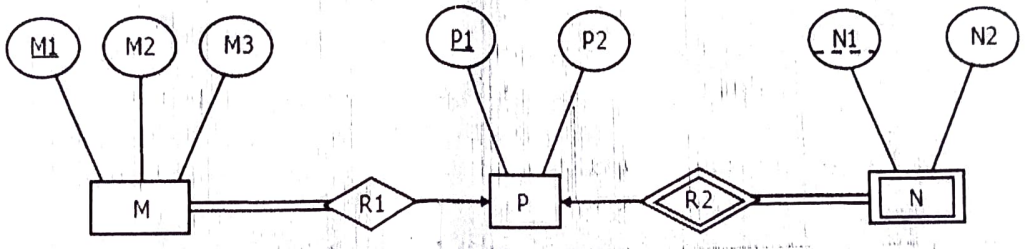
Minor Test

MCAE403: Database Applications

Time : 1 hr

Max Marks : 20

Q.1: Consider the following ER diagram:



The minimum number of tables needed to represent M, N, P, R1, R2 is _____ and the tables created would be _____.

Justify your answer. (2)

Q.2: The functional dependencies for relation R(ABCDE) are

{A->BC, CD->E, B->D, E->A}

(a) Find the highest normal form the R is in. (1)

(b) Convert the relation into the highest normal form (BCNF). (2)

(c) Check whether the decomposition is lossless and functional dependency preserving. (2)

Q.3: Differentiate between UNION and UNION ALL operators. Write two conditions which have to be fulfilled if you want to use INTERSECT operator. (2)

Q.4: Refer the relation given below: (4)

City(City_id: serial, **city**: varchar(50), **country_id**: int, **last_update**: timestamp)

a) Find the difference between the total number of CITY entries in the table and the number of distinct CITY entries in the table.
b) Find all the cities with even city_id.
c) Query the list of CITY names ending with vowels (i.e. a,e,i,o,u). Your result should not contain duplicate values.
d) Query the list of CITY names from the CITY table which have vowels (i.e., a, e, i, o, and u) as both their first and last characters. Your result cannot contain duplicates.

Q.5: Write SQL queries to do the following: **(4)**

a) Find the total number of products in each category. Sort the result by the total number of products, in descending order.

> **Product**: **product_id**:smallint, **product_name**:varchar, **supplier_id**:smallint, **category_id**:smallint, **quantity_per_unit**:varchar(20), **unit_price**:real, **units_in_stock**:smallint, **units_on_order**:smallint, **reorder_level**:smallint, **discontinued**:integer
>
> **Categories: category_id:** smallint, **category_name:** varchar(15), **description**:text, **picture**:bytea

b) Find the total number of customers per country and city.

> Customers:
> **customer_id:** smallint, **company_name:**varchar(50), **contact_name:**varchar(50), **contact_title:**varchar(50), **address:**varchar(50), **city:**varchar(50), **region:**varchar(50), **postal_code:**varchar(50), **country,phone:**varchar(50), **fax:**varchar(50)

c) A salesperson for NorthWind is going on a business trip to visit customers, and would like to see a list of all customers, sorted by region, alphabetically. However, he wants the customers with no region (null in the region field) to be at the end, instead of at the top. Within the same region, companies should be sorted by CustomerID. Refer table **Customers.**

d) Show a list of countries where the NorthWind company has customers. Refer table **Customers.**

Q.6: Write a query to show how the average order amount for each store compares to the minimum and maximum of the average order amount of all stores. (3)

> **orders_store(Id**: int8, **date**: date, **customer_id**: int8, **store**: varchar, **employee_id**: int8, **amount**: numeric)

	store	average_order	min_avg_order	max_avg_order
0	Center	489.666667	338.5	725.0
1	East	338.500000	338.5	725.0
2	West	725.000000	338.5	725.0