DBMS HOMEWORK-03

1. Draw an ER diagram of restaurant order management application. The database consists of several entities:

- Restaurant: Relevant attributes are **restaurant_id**, restaurant_name, location
- Employee: Relevant attributes are **employee_id**, name, Job_role
- Customer: All customers have some shared attributes: **customer_id**, name, phone_number. For online customers who registered on the website, they have extra attributes: Email address and password
- Dish: Relevant attributes are **item_id**, name, description, price
- Order: Relevant attributes are **order_id**, **restaurant_id** , customer_id, item_id, timestamp, comment

Where key of each table is marked in bold text. Also consider following constrains:

a) There are two types of customers: eat in customers and others who registered online, you should use the **ISA structure** to describe the structure of those customers.

b) Each restaurant would have multiple employees, and each employee can only be employed by one restaurant, and for each restaurant, there will be **only one staff member who is the manager of that restaurant.**

c) Each order record only contains one dish that's been ordered, each customer can give multiple orders, one of the employees will take the order and process it.

Solution:

a. **Entities and Attributes:**
   - Restaurant: Restaurant_id(PK), Restaurant_name,Location,,Manager_id(FK).
   - Employee: Employee_id(PK), Employee_name, Job_Role,Restaurant_ID(FK).
   - Manager: Manager_id(PK), Restaurant_id(FK),Employee_id(FK).
   - Customer: Customer_id(PK),Customer_name,Customer_type.
   - EatinCustomer: EatIn_id(PK),Customer_name,Phone_number,Customer_Id(FK).
   - OnlineCustomer: Online_id(PK),Customer_name,Email,password,Phone_number, Address,Customer_id(FK).
   - Menu: Item_id(PK),Item_Name,Description,Cost,Restaurant_id(FK).
   - Order: Order_ID(PK), Quantity,Instructions,Timestamp,Item_id(FK),Restaurant_id(FK).

b. **Relationships:**
   - Employee works at the Restaurant. (N:1)
   - Manager manages Restaurant. (1:1)
   - Manager manages Employees. (1: N)
   - Employee handles orders. (1: N)
   - Orders are placed in the Restaurant. (N:1)
   - The restaurant has a menu. (N:1)
   - Orders are placed from the menu. (N:1)
   - Customer places the order. (1:N)

Figure: ER Model

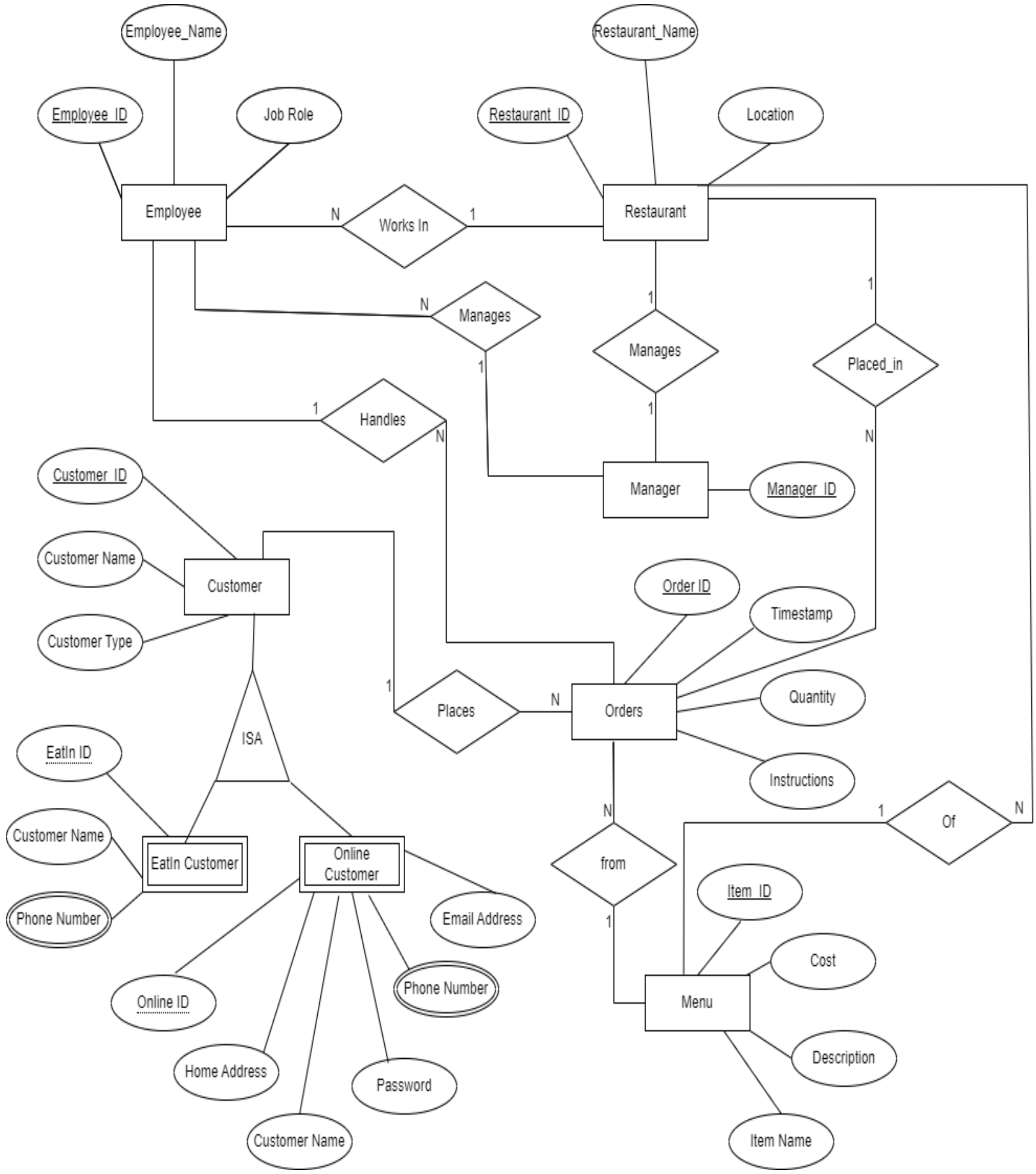
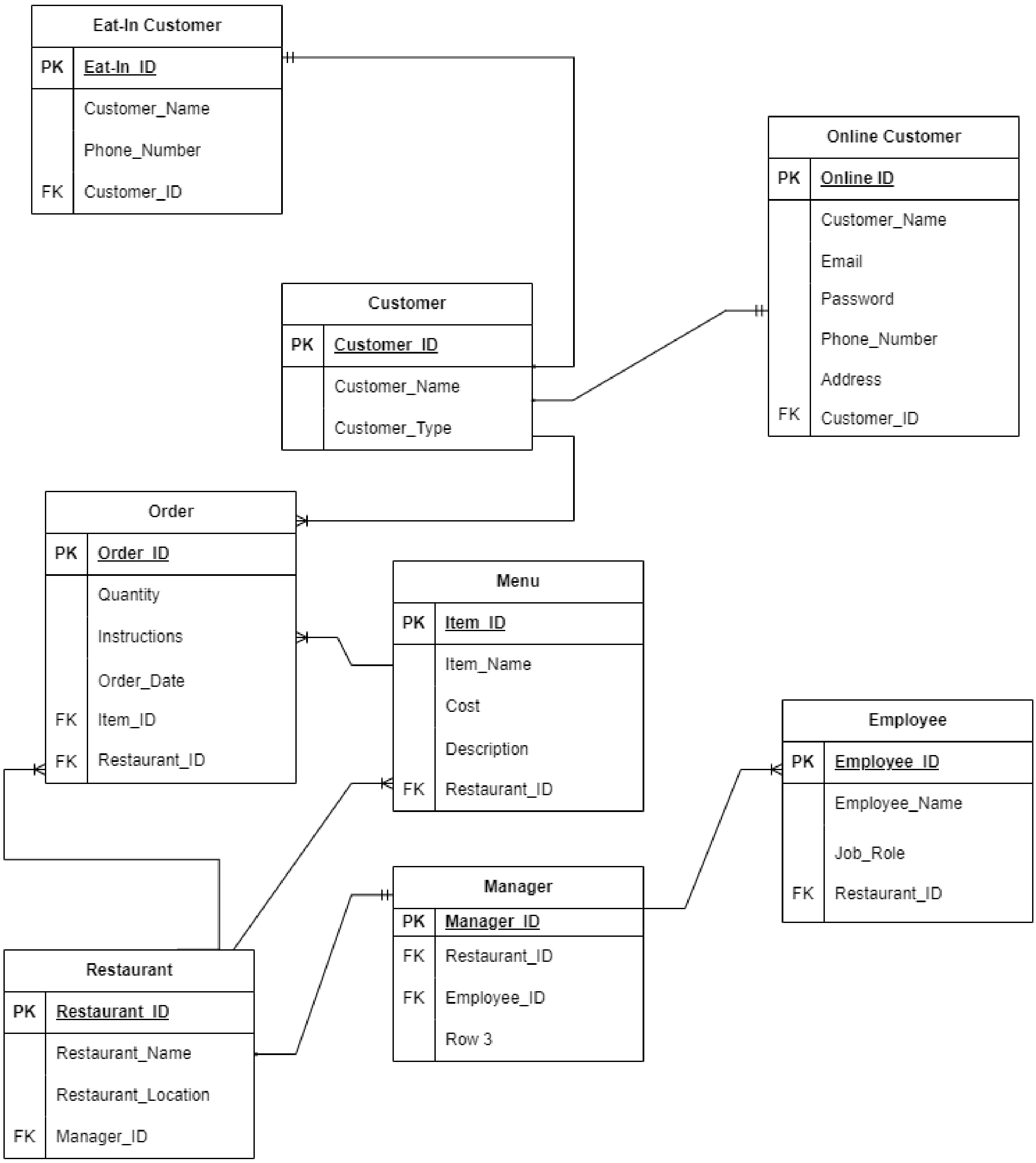


Fig: Relational Schema



2. Translate the ER diagram from Q1 into SQL DDL statements.

Solution:

Restaurant table

```
create table Restaurant (
Restaurant_id int primary key,
Restaurant_name varchar (255),
Location varchar (255),
Manager_id int,
foreign key (Manager_id) references Manager(Manager_id)
);
```

Employee table

```
create table Employee (
Employee_id int primary key,
Employee_name varchar (255),
Job_Role varchar (255),
Restaurant_id int,
foreign key (Restaurant_id) references Restaurant (Restaurant_id)
);
```

Manager table

```
create table Manager (
Manager_id int primary key,
Restaurant_id int,
Employee_id int,
foreign key (Restaurant_id) references Restaurant (Restaurant_id),
foreign key (Employee_id) references Employee (Employee_id)
);
```

Customer table

```
create table Customer (
Customer_id int primary key,
Customer_name varchar (255),
Customer_type varchar (255)
);
```

Eat-In Customer table.

```
create table EatinCustomer (
EatIn_id int primary key,
Customer_name varchar (255),
Phone_number varchar(15),
Customer_id  int,
foreign key(Customer_id) references Customer(Customer_id)
);
```

Online Customer table

```
create table OnlineCustomer (
Online_id int primary key,
Customer_name varchar(255),
Email varchar(255),
Password varchar(255),
Phone_number varchar(15),
Address varchar (255),
Customer_id int,
Foreign key(Customer_id) references Customer(Customer_id)
);
```

Menu table

```
create table Menu (
Item_id int primary key,
Item_Name varchar (255),
Description text,
Cost decimal (10, 2),
Restaurant_id int,
Foreign key (Restaurant_id) references Restaurant (Restaurant_id)
);
```

Order table

```
create table Order (
Order_ID int primary key,
Quantity int,
Instructions text,
Order date timestamp,
Item_id int,
Restaurant_id int,
Foreign key (Item_id) references Menu (Item_id),
Foreign key (Restaurant_id) references Restaurant (Restaurant_id)
);
```

3. Consider Table (a) which **shows part of the records** in relation R. Complete the table (b) for given functional dependencies (FD). Please just answer yes, no or unknown

A	B	C	D
A_1	B_1	C_1	D_1
A_1	B_1	C_2	D_2
A_1	B_2	C_1	D_1
A_3	B_3	C_2	D_2
A_4	B_3	C_3	D_3
A_4	B_1	C_4	D_1
A_5	B_4	C_3	D_1
A_5	B_3	C_2	D_2
A_3	B_4	C_3	D_1

Table (a)

FD	Satisfied on given records. (yes/ no/unknown)	Hold on R (yes/no/unknown)	Trivial (yes/no)
A→ B	No	No	No
B→ A	No	No	No
AC→ D	Yes	Yes	No
AC→B	No	No	No
AD→ B	No	No	No
ABC→ AC	Yes	Yes	Yes
BC→ D	Yes	Yes	No
C→ AC	No	No	Yes
AB→ A	Yes	Yes	Yes
BD→ A	No	No	No

Table (b22)

4. Consider a relation R1(A,B,C,D,E,F,G) and a set of functional dependencies FD = {AB → E, C → D, DE → F, DA → G} which hold on R1. Using Armstrong's axioms verify if the following functional dependencies hold on R1

FD	Yes/No	Proof if yes
AC → G	Yes	AC→AC (reflexivity) C→D and DA→G(given) If DA→G, D→G and A→G(augmentation) So (AC)+ = {A, C, D, G} Hence AC->G
AC→ F	No	
ABD→F	Yes	ABD→ ABD (reflexivity) AB→E, C→D, DE→F and DA→G(given) So (ABD)+ = {A, B, D, C, E, F, G} (transitivity and augmentation)
BCD→ F	No	
ABC→EG	Yes	ABC→ ABC (reflexivity) AB→E, C→D, DE→F and DA→G(given) So (ABC)+ = {A, B, C, D, E, F, G} (transitivity and augmentation)

5. Consider a relation R(A,B,C,D,E,F) and a set of functional dependencies, which hold on R : { CD->BE, B->CA, D->F} Are decompositions in the table lossless and why?

Decomposition	Lossless? (Yes) /(No)	Why
R1(ABF) and R2(CDE)	No	R1 and R2 do not have any common attributes.
R1(ABCEF) and R2(CDE)	No	R = R1 U R2. CE is a common attribute and CE determines only CE and does not determine A, B, D, F.
R1(ABDE) and R2(BDF)	No	R1 U R2 does not give R
R1(ACDF) and R2(BCDE)	Yes	R = R1 U R2. CD is a common attribute and CD → {A, B, C, D, E, F} (transitive and reflexivity)
R1(ABEF) and R2(BCDF)	Yes	R = R1 U R2. BF is a common attribute and BF → {A, B, C, D, E, F} (transitive and reflexivity)

6. Consider the following relations with the associated functional dependencies. Decide, whether those relations are in (a) BCNF, (b)3NF, (c) neither in BCNF nor 3NF normal form.

Relation, FD	Answer (a, b, or c)	Solution
R1(A,B,C,D) {AB→ C, C→ D}	(c) Neither	determinant (C) of the FD C→ D is not a super key, hence it is not in 3NF and C does not determine {A,B}, hence not in BCNF
R2(A,B,C,D), {AC→ BD}	(a) BCNF (b) 3NF	(AC)+ → {A,B,C,D} The determinant (AC) of the FD AC→ BD is a super key, and it is also in 3NF because every non-prime attribute is either a super key or part of a key.
R3(A,B,C,D) {AB→ CD, D→A}	(b)3NF	AB is the candidate key and A in FD D→ A is a prime attribute hence it is 3NF and D does not determine {B, C}, hence not in BCNF
R4(A,B,C,D,E), {AC→D, D→B}	(c) Neither	determinant (D) of the FD D→B is not a super key hence not in 3NF and AC does not determine {E} and D does not determine {A, C,E} hence not in BCNF
R5(A,B,C,D,E) {CD→ ABE, B→C}	(b)3NF	CD is a candidate key and C in FD B→C is a prime attribute hence 3NF and B does not determine {A,D,E} hence not in BCNF

7. Table T in your database D has size 4 kBytes. How much space does Table T take on the drive? Show your calculations.

Solution:

Given:
Disk Block Size: 5kbytes
1kbytes: 1024 bytes
1Mbytes: 1024kbytes

Table size = 4*1024=4096 bytes
Block size = 5*1024 = 5120 bytes
Space taken = 4096/5120 = 0.8 block

Table T takes up 0.8 of a block.

8. Table T in your database D has size 400 MBytes. You execute a query: “select * from T”. How much data will be read from the drive? Assume that n*size_of_tuple = block_size, where n is a natural number. Show your calculations.

Solution:

Given:
Disk Block Size: 5kbytes
1kbytes: 1024 bytes
1Mbytes: 1024kbytes

n*size_of_tuple = block_size
n*size_of_tuple = 5kbytes
n=5kbytes/size_of_tuple

Table size = 400*1024k = 409600kbytes
Disk Block size = 5kbytes
No. of blocks= Table size/Disk Block size = 409600kbytes/5kbytes = 81920bytes
Data read = Number of blocks occupied by Table T * Block size= 81920bytes*5kbytes = 409600kbytes ≈ 400Mbytes

9. Assume that table T is defined in the same way as in question 8. You execute a query “select * from T where num=300”. How many blocks and how many bytes will be read from the disk? Show your calculations (There is no index built on this column).

Solution:

Given:
Disk Block Size: 5kbytes
1kbytes: 1024 bytes
1Mbytes: 1024kbytes

Table size = 400*1024k = 409600kbytes

Disk Block size = 5kbytes

No. of blocks= Table size/Disk Block size = 409600kbytes/5kbytes = 81920bytes

Since there is no index built on the column 'num', the entire table will be scanned to find the rows where 'num=300

Data read = Number of blocks occupied by Table T * Block size= 81920*5kbytes = 409600kbytes ≈ 400Mbytes.

10. How your answer will be different from question 9, if clustered index for column num is used for table T. Assume that size of the required index structure is 8 kBytes and there are m records with num = 300, which may be stored in n different blocks.

Solution:

Given:
Disk Block Size: 5kbytes
1kbytes: 1024 bytes
1Mbytes: 1024kbytes

size of index structure = 8 kBytes = 8*1024=8192bytes
Number of blocks to read = 8192bytes/5*1024bytes = 1.6
As data is stored in n different blocks therefore the total number of blocks is n * 1.6 blocks.
Data read = Number of blocks occupied by Table T * Block size= 1.6*5kbytes = 8192kbytes ≈ 8kbytes.

11. Draw a valid B+ tree below for the search keys (1, 2, 3, 4, …, 12). Assume the keys are inserted in their natural order. The order of the tree is 3

Solution: