

AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Faculty of Science and Technology



Project Cover Page

Assignment Title:	Database Project		
Assignment No:	01	Date of Submission:	12 December 2021
Course Title:	INTRODUCTION TO DATABASE		
Course Code:	CSC2108	Section:	J
Semester:	Fall 2021-22	Course Teacher:	ANJIR AHMED CHOWDHURY

Declaration and Statement of Authorship:

1. I/we hold a copy of this Assignment/Case-Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case-Study is my/our original work and no part of it has been copied from any other student's work or from any other source except where due acknowledgement is made.
3. No part of this Assignment/Case-Study has been written for me/us by any other person except where such collaboration has been authorized by the concerned teacher and is clearly acknowledged in the assignment.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared and archived for the purpose of detecting plagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by the Faculty for review and comparison, including review by external examiners.
7. I/we understand that Plagiarism is the presentation of the work, idea or creation of another person as though it is your

Faculty use only

FACULTY COMMENTS	Marks Obtained	
	Total Marks	

own. It is a form of cheating and is a very serious academic offence that may lead to expulsion from the University. Plagiarized material can be drawn from, and presented in, written, graphic and visual form, including electronic data, and oral presentations. Plagiarism occurs when the origin of them material used is not appropriately cited.

8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or to copy my/our work.

* Student(s) must complete all details except the faculty use part.

** Please submit all assignments to your course teacher or the office of the concerned teacher.

No	Name	ID	Program
1	UMA CHAKRABORTY	20-43892-2	BSc [CSE]
2	ADITYA KUMAR SAHA	20-43958-2	BSc [CSE]

Introduction

This project is about McDonald's Management System. McDonald's is the worldwide largest restaurant franchise. In this project, we stored a database of employees, branches, orders, billing information etc. We stored every information in some tables so that it can be analyzed and also it can be updated when it is needed.

Tables

Restaurant:

R_NAME	R_CONTACT	ADDRESS
McDonalds LA	8812	Groove Street
McDonalds Vegas	8621	Craig Road
McDonalds Texas	8845	Killeen
McDonalds California	8832	Bridge City
McDonalds NY	8968	Herald Square

Cashier:

C_ID	C_NAME
101	Uma
102	Jotirmoyee
103	Amit
104	Achintya
105	Anik

Bill:

B_NO	PRICE	OTHER_DETAILS	C_ID
1	240.25	Parcel	101
2	700	Table B	102
3	1020	Parcel	103
4	1300.22	Table K	104
5	360	Table AA	105

Customer:

C_NAME	C_ADDRESS	C_CONTACT	C_ID
Aaron Finch	Santa Monica	1683821144	801
David Warner	Gotham	1783820051	802
Steven Smith	Arcadia	1523821184	803
Uma Chakraborty	El Monte	1518452034	804
Achintya Mukherjee	El Monte	1713902476	805
Sanchita Saha	Paramount	1863783009	806
Thomas Muller	Lancaster	1718716804	807
Manuel Neuer	Munich	1683654755	808
Adrita Jahan	Elma	1983851152	809
Pat Cummins	Lynnwood	1692821044	810

Item:

I_NO	PRICE	QUANTITY
1	230	3
2	500	2
3	190.5	3
4	280.35	1
5	150	2
6	800	7
7	620	2
8	130	9
9	990	6
10	730	1

Manager:

M_NAME	M_ID	M_CONTACT
Richardson	201	1313902485
Charlie	202	1613903485
Sanchita	203	1313900195
Jotirmoyee	204	1810902085
Krishna	205	1388902995

Orders:

O_ID	O_DATE	O_TYPE
1	01-NOV-21	Counter
2	03-NOV-21	Counter
3	05-NOV-21	Online Order
4	06-NOV-21	Dine In
5	12-NOV-21	Counter
6	18-NOV-21	Dine In
7	25-NOV-21	Counter
8	30-NOV-21	Online Order
9	01-DEC-21	Dine In
10	02-NOV-21	Dine In

Normalization 1NF

Unique Column Name:

All the names of columns of a table must be unique. As we can see in our Orders table, the column names are O_ID, O_Date and O_type. Every name is unique and doesn't match with other tables. But when we created the table it was named as ID, Date and Type. Then we make it unique names.

Single Value:

All the columns must store a single value. We can see it as an example of our order table. When we created this table it didn't have a single stored value. At the O_Type column it had Counter, Dine In at number 1 row. But after normalization we make it a single stored column.

Same Domain:

All the values of the rows must be in the same domain. For example, at the manager table all the names of the managers are stored in only the M_Name column. We cannot use other values

except the name in that column. After normalization we corrected this error of all the tables.

Order:

There is no restriction for orders in stored data of the table. As an example, we can see in the Item table at the Quantity column the order is not maintained.

Normalization 2NF

All tables of our project are already normalized 1NF. Now no tables have the partial dependency. So all the tables follow the 2NF rules.

Normalization 3NF

As the tables are already normalized in 2NF and there is no transitive dependency. So all the tables follow the 3NF rules.

ER Diagram

Case Study:

McDonald's is one of the most popular and widely spread fast-food restaurants in our country. In the McDonald's management system the cashier's id, name will be stored in the database. The restaurant has a cashier to manage the system of McDonald's. There are a lot of branches of McDonald's restaurants. So, in the restaurant where the restaurant's name, multiple contact number, the address will be stored in the database. The restaurant is managed by a number of managers. The information of the manager (Name, ID number, contact number) will be recorded in the database. The manager takes orders. The information about the order (Order number, Order Date, Order Type) will be stored. The information of the item (Quantity, Item Number, Price, and Description) will be stored in the database. The order contains the item. The order places the customer and the information of the customer are (Name, address, contact number, id). The customer pays the bill. Here the information about the bill (Bill number, price, and other details) will be recorded. The bill was paid to the cashier. All the above information will be stored in the database. Now drawing an ER Diagram according to the mentioned scenario.

ER Diagram:

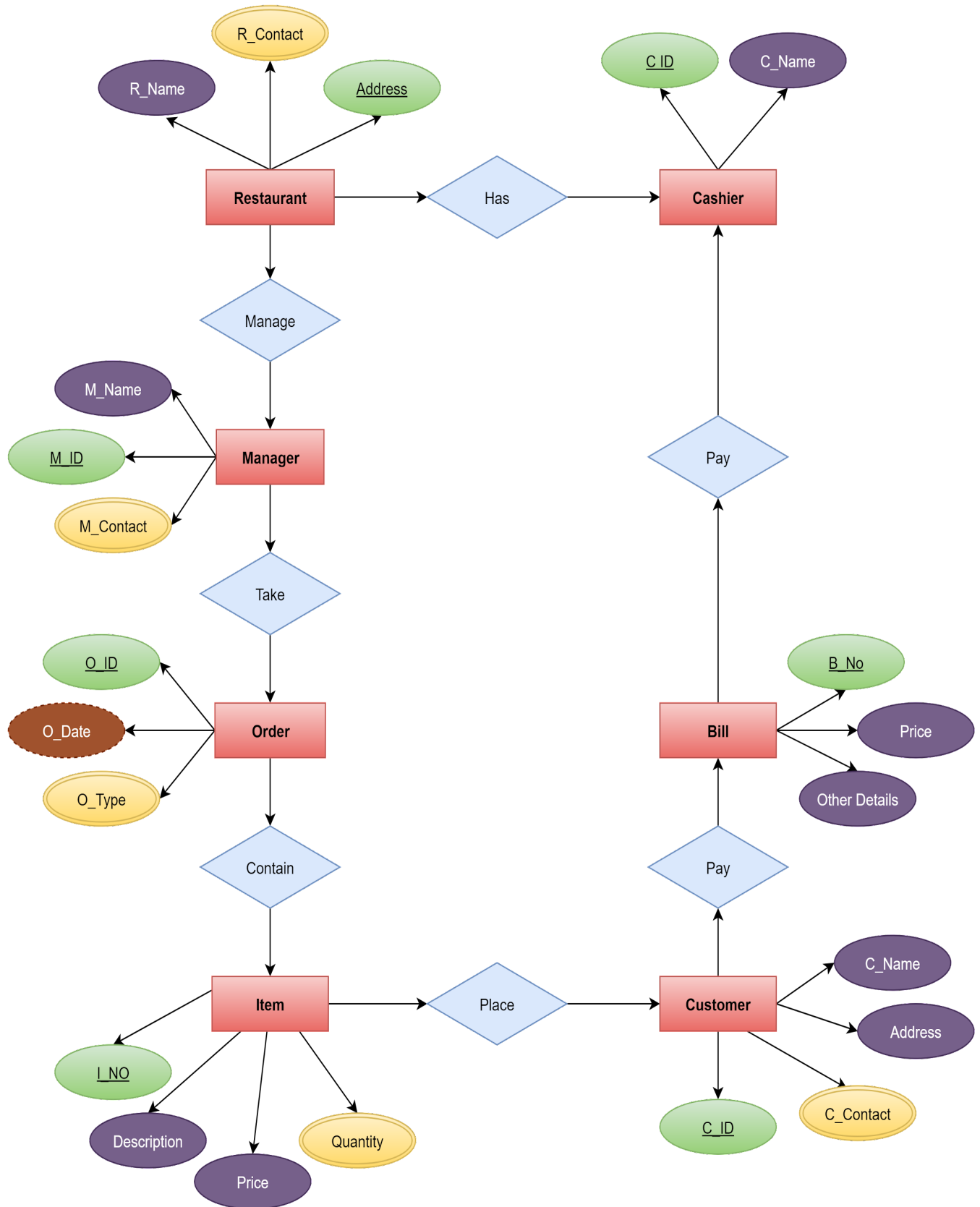


Table Creation (DDL)

Cashier:

```
create table Cashier
(
  c_id number(10) constraint cashier_cid_pk primary key,
  c_name varchar2(20)
)
```

Restaurant:

```
create table Restaurant
(
  r_name varchar2(30),
  r_contact number(20),
  address varchar2(30) constraint Restaurant_address_pk primary key
)
```

Bill:

```
create table bill
(
  b_no number(20) constraint bill_bno_pk primary key,
  price number(10,2),
  other_details varchar2(20),
  c_id number(10) constraint bill_cid_fk references Cashier(c_id)
)
```


Customer:

```
create table customer
(
  c_name varchar2(20),
  c_address varchar2(20),
  c_contact number(20),
  c_id number(10) constraint customer_cid_pk primary key
)
```

Orders:

```
create table orders
(
  o_id number(10) constraint orders_oid_pk primary key,
  o_date date,
  o_type varchar2(20)
)
```

Item:

```
create table item
(
  i_no number(10) constraint item_ino_pk primary key,
  price number(10,2),
  quantity number(10)
)
```

Manager:

```
create table manager
(
  m_name varchar2(20),
  m_id number(10) constraint manager_mid_pk primary key,
  m_contact number(20))
```

Data Insertion (DML)

Cashier: (**Sequence**)

```
CREATE SEQUENCE cash
INCREMENT BY 1
START WITH 101
MAXVALUE 105
NOCACHE
NOCYCLE;
```

```
insert into cashier (c_id,c_name) values (cash.nextval,'Uma')
insert into cashier (c_id,c_name) values (cash.nextval,'Jotirmoyee')
insert into cashier (c_id,c_name) values (cash.nextval,'Amit')
insert into cashier (c_id,c_name) values (cash.nextval,'Achintya')
insert into cashier (c_id,c_name) values (cash.nextval,'Anik')
```

Restaurant:

```
insert into restaurant (r_name,r_contact,address) values ('McDonalds
LA',08812,'Groove Street')
insert into restaurant (r_name,r_contact,address) values ('McDonalds
Vegas',08621,'Craig Road')
insert into restaurant (r_name,r_contact,address) values ('McDonalds
Texas',08845,'Killeen')
insert into restaurant (r_name,r_contact,address) values ('McDonalds
California',08832,'Bridge City')
insert into restaurant (r_name,r_contact,address) values ('McDonalds
NY',08968,'Herald Square')
```

Bill:

```
insert into bill(b_no,price,other_details,c_id)
values(1,240.25,'Parcel',101)
insert into bill(b_no,price,other_details,c_id) values(2,700,'Table
B',102)
insert into bill(b_no,price,other_details,c_id)
values(3,1020,'Parcel',103)
insert into bill(b_no,price,other_details,c_id)
values(4,1300.22,'Table K',104)
insert into bill(b_no,price,other_details,c_id) values(5,360,'Table
AA',105)
```

Customer:

```
insert into customer (c_id,c_name,c_address,c_contact) values
(801,'Aaron Finch','Santa Monica',1683821144)

insert into customer (c_id,c_name,c_address,c_contact) values
(802,'David Warner','Gotham',1783820051)

insert into customer (c_id,c_name,c_address,c_contact) values
(803,'Steven Smith','Arcadia',1523821184)

insert into customer (c_id,c_name,c_address,c_contact) values
(804,'Uma Chakraborty','El Monte',1518452034)

insert into customer (c_id,c_name,c_address,c_contact) values
(805,'Achintya Mukherjee','El Monte',1713902476)

insert into customer (c_id,c_name,c_address,c_contact) values
(806,'Sanchita Saha','Paramount',1863783009)

insert into customer (c_id,c_name,c_address,c_contact) values
(807,'Thomas Muller','Lancaster',1718716804)

insert into customer (c_id,c_name,c_address,c_contact) values
(808,'Manuel Neuer','Munich',1683654755)
```

```
insert into customer (c_id,c_name,c_address,c_contact) values
(809,'Adrita Jahan','Elma',1983851152)
```

```
insert into customer (c_id,c_name,c_address,c_contact) values
(810,'Pat Cummins','Lynnwood',1692821044)
```

Item:

```
insert into item(i_no,price,quantity) values(1,230.00,3)
insert into item(i_no,price,quantity) values(2,500.00,2)
insert into item(i_no,price,quantity) values(3,190.50,3)
insert into item(i_no,price,quantity) values(4,280.35,1)
insert into item(i_no,price,quantity) values(5,150.00,2)
insert into item(i_no,price,quantity) values(6,800.00,7)
insert into item(i_no,price,quantity) values(7,620.00,2)
insert into item(i_no,price,quantity) values(8,130.00,9)
insert into item(i_no,price,quantity) values(9,990.00,6)
insert into item(i_no,price,quantity) values(10,730.00,1)
```

Orders:

```
insert into orders(o_id,o_date,o_type) values(1,'1
November,2021','Counter')
insert into orders(o_id,o_date,o_type) values(2,'3
November,2021','Counter')
insert into orders(o_id,o_date,o_type) values(3,'5
November,2021','Online Order')
insert into orders(o_id,o_date,o_type) values(4,'6
November,2021','Dine In')
insert into orders(o_id,o_date,o_type) values(5,'12
November,2021','Counter')
insert into orders(o_id,o_date,o_type) values(6,'18
November,2021','Dine In')
insert into orders(o_id,o_date,o_type) values(7,'25
November,2021','Counter')
```

```
insert into orders(o_id,o_date,o_type) values(8,'30  
November,2021','Online Order')
```

Manager:

```
insert into manager (m_name,m_id,m_contact)  
values('Richardson',201,1313902485)  
insert into manager (m_name,m_id,m_contact)  
values('Charlie',202,1613903485)  
insert into manager (m_name,m_id,m_contact)  
values('Sanchita',203,1313900195)  
insert into manager (m_name,m_id,m_contact)  
values('Jotirmoyee',204,1810902085)  
insert into manager (m_name,m_id,m_contact)  
values('Krishna',205,1388902995)
```

Query

1. Display all the records of orders which are taken for 'Dine In'

```
select * from orders where O_Type='Dine In'
```

Relational Algebra:

$$\sigma_{o_type = \text{"Dine In"}} \text{ orders}$$

2. Display the customer names who are living at El Monte.

```
select C_Name from customer where C_Address='El Monte'
```

Relational Algebra:

$$\pi_{c_name} \sigma_{c_address = \text{"El Monte"}} \text{ customer}$$

3. Increase the 1.5 times higher price of the item table and display all the records.

```
select I_NO, price, quantity, price * 1.5 "Updated Price" from item
```

Relational Algebra:

```
 $\Pi$  i_no, price, quantity, price * 1.5 item
```

4. Display all the orders after 30 November, 2021.

```
select * from orders where O_Date > '30-NOV-21'
```

Relational Algebra:

```
 $\sigma$  o_date > "30-NOV-21" orders
```

5. Concatenate Customer name & address and display as Customer Details Column.

```
select C_Name || ' ' || 'lives in' || ' ' || C_Address AS "Customer Details"  
from customer
```

6. Display the cashier name whose name starts with A.

```
select C_Name from cashier where C_Name like 'A%'
```

Relational Algebra:

```
 $\Pi$  c_name  
 $\sigma$  c_name LIKE "A%" cashier
```

7. Write a query and display the total price of food which is ordered for parcel.

```
select sum(price) from bill where other_details='Parcel'
```

Relational Algebra:

```
 $\Pi$  SUM (price)
 $\gamma$  SUM (price)
 $\sigma$  other_details = "Parcel" bill
```

8. Write a query and display the average price of the item where quantity is equal or more than three.

```
select avg(price) from item where quantity >= 3
```

Relational Algebra:

```
 $\Pi$  AVG (price)
 $\gamma$  AVG (price)
 $\sigma$  quantity >= 3 item
```

9. Display the round value of the average price of the bill.

```
select round(avg(price)) from bill
```

10. Display the customer name and contact who lives in Paramount and id is 806.

```
select C_Name, C_Contact from customer where C_Address= 'Paramount'
and C_ID=806
```

Relational Algebra:

```
 $\Pi$  c_name, c_contact
 $\sigma$  c_address = "Paramount" AND c_id = 806 customer
```

11. Display all the names of the restaurant except which is at Herald Square.

```
select R_Name from restaurant where address not in('Herald Square')
```

Relational Algebra:

```
 $\Pi$  r_name  
 $\sigma$  NOT (address = "Herald Square") restaurant
```

Sub-query

12. Write a query and display Other details, price and customer id whose name starts with 'J'.

```
select Other_Details, price, c_id from bill where c_id=(select c_id  
from cashier where c_name like 'J%')
```

13. Write a query and display bill no, price, other details which id is 102.

```
select B_NO, price, Other_Details from bill where c_id=(select c_id  
from cashier where c_id=102)
```

14. Show maximum price of first five items.

```
select max(price) from item where I_No<(select I_No from item where  
I_No like '6')
```

15. Display price of item whose order id is 3.

```
select price from item where I_NO=(select O_ID from orders where  
O_ID=3)
```


Joining

16. Write a query and display order date, order type, price and quantity. (Equijoin)

```
select o.o_date, o.o_type, i.price, i.quantity from orders o,item i
where o.o_id=i.i_no
```

17. Write a query and display all the customer name, customer address, customer contact and bill no. (Outer Join)

```
select c.c_name, c.c_address, c.c_contact, b.b_no from customer c,
bill b where c.c_id=b.c_id(+)
```

18. Write a query and display the restaurant names and addresses by joining them. (Self Join)

```
select r1.r_name || ' is in ' ||r2.address as "Restaurant Details"
from restaurant r1, restaurant r2 where r1.r_name=r2.r_name
```

View

19. Create a view to display where customers live.

```
create view aview as
select c1.c_name || ' from ' ||c2.c_address as "Customer Details"
from customer c1, customer c2 where c1.c_name=c2.c_name
select * from aview
```

20. Create a view to display order date, order type, price and quantity.

```
create view bview as
select o.o_date, o.o_type, i.price, i.quantity from orders o,item i
where o.o_id=i.i_no
select * from bview
```

Contribution

Name	Topic
ADITYA KUMAR SAHA (20-43958-2)	Introduction, Case Study, DDL, Sequence, Sub-Query, Joining, View
UMA CHAKRABORTY (20-43892-2)	File formatting, Normalization, ER Diagram, DML, Query, Relational Algebra