Restaurant Management System

Submitted to
Nashia Ahmed Nabila

By
Nayan, Md. Tanzirul Haque (16-31332-1)
Hossain, Md. Sabbir (16-31709-1)
Hossain, Sakib (16-31288-1)
Hossain, Mohammad Al Amin (16-31286-1)

Introduction to Database (CSC 2208)
American International University – Bangladesh (AIUB)

Table of Contents

Business and System Summary

Customer satisfaction is the key to success for any business. In a restaurant, the traditional hand-waving method for calling services is inefficient often leading to many complaints. The Restaurant Management System increases operational efficiency through use of a database. It is easy and effective managing a restaurant by using a database. A restaurant database contains customer informations, employee informations, food item details, order details and essential data. By using a database, restaurant manager can easily analyze previous data and come to a decision in a few seconds which ensures a proper management. By using a database, customer orders can be served on time in a perfect way which ensures customer satisfaction. A restaurant database also ensures a better management system as well as customer satisfaction. So, a restaurant database is really essential for a proper management.

Overview of the Business Environment and Project Objectives

In many popular restaurants, waiters/waitresses tend to miss out on tables or customers' calls during busy hours potentially decreasing ones clientele as well as the traditional hand-waving method for calling services is inefficient often leading to many complaints. While this is an ongoing issue, there is still no product that drastically improves the communication between the servers and the customers in the current market. Hence, the goal is to design a system in which the customers can call their servers easily and help the restaurant increase overall efficiency. It is easy and effective managing a restaurant by using a database. A restaurant database contains customer informations, employee informations, food item details, order details and essential data. By using a database, restaurant manager can easily analyze previous data and come to a decision in a few seconds which ensures a proper management. This database for restaurant management can monitor the work-flow of any employee of the restaurant which ensures the proper management of the restaurant. The objective of the project is to enhance efficiency and to manage the restaurant in a better and easier way.

Brief Technical Summarization of Developed Database

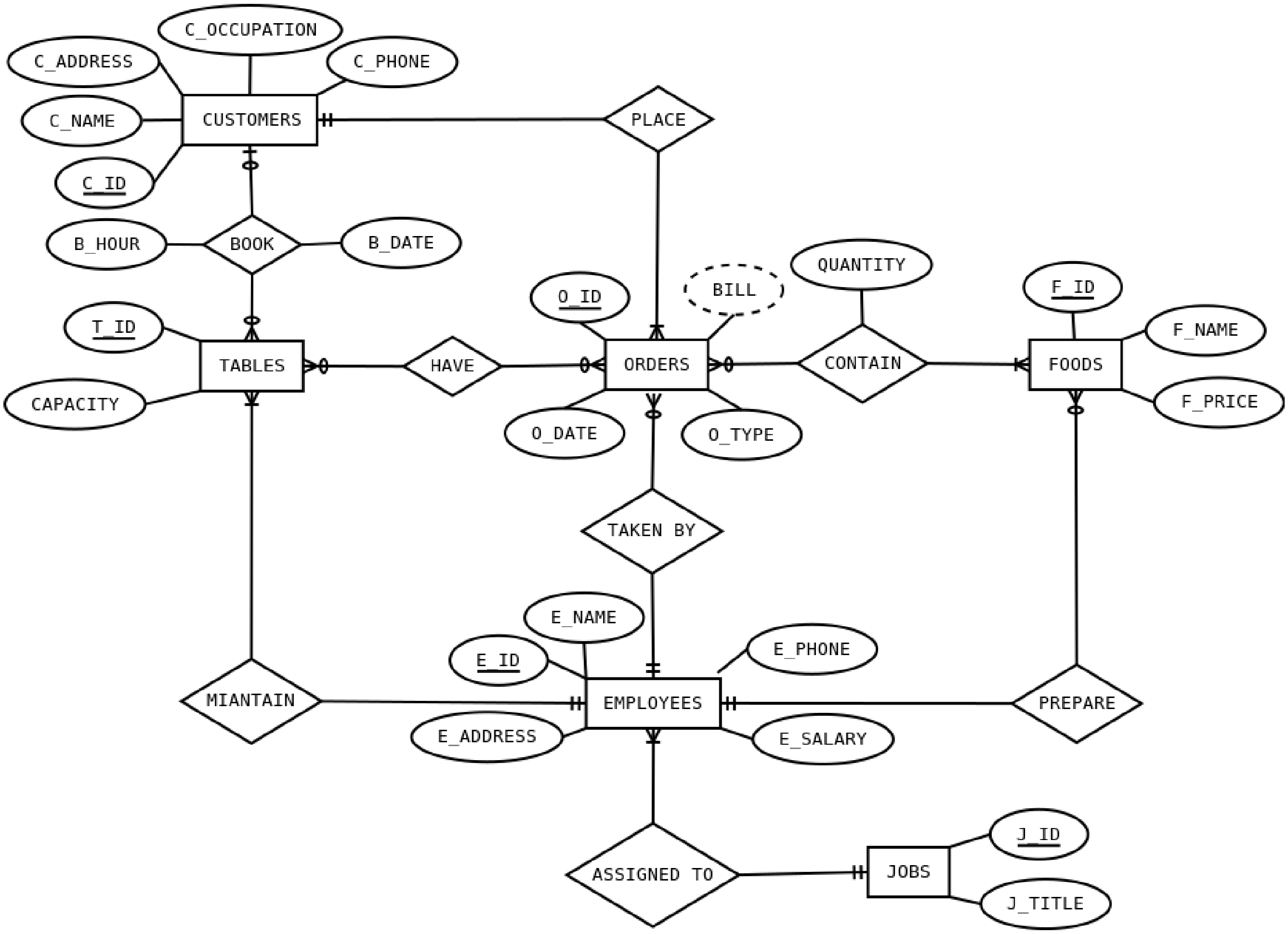
First, an Entity Relationship Diagram was developed. Then tables and attributes were derived by using normalization process. After that, a schema of the database was drawn and then the database was developed. Developed database contains total 9 tables which are BOOKING, JOBS, CUSTOMERS, EMPLOYEES, TABLES, ITEMS, ORDERS, ORDER_HISTORY. The BOOKING table contains two foreign key named C_ID , T_ID from CUSTOMERS and TABLES tables, booking date as B_DATE attribute and lastly booking hour as B_HOUR attribute. The JOBS table contains a primary key called J_ID which represents job id, job title as J_TITLE attribute. The CUSTOMERS table contains one primary key named C_ID which represents customer id, customer's name as C_NAME attribute, customer's address as C_ADDRESS attribute, customer's phone number as C_PHONE, customer's occupation as C_OCCUPATION attribute. The EMPLOYEES table contains one primary key E_ID which is employee's id, employee's name as E_NAME attribute, employee's phone number as E_PHONE attribute, employee's address as E_ADDRESS attribute, employee's salary as E_SALARY attribute and a foreign key named J_ID from JOBS table. TABLES table contains a primary key named T_ID which represents table id, table capacity as CAPACITY attribute and a foreign key named E_ID from EMPLOYEES table. FOODS table contains a primary key named F_ID which represents food id, food price as F_PRICE attribute, food name as F_NAME attribute and a foreign key named E_ID from

EMPLOYEES table. ITEMS table contains quantity of ordered foods as QUANTITY attribute, two foreign keys named O_ID from ORDERS table and F_ID from FOODS table. ORDERS table contains a primary key named O_ID which represents order id, order type as O_TYPE attribute, order date as O_DATE attribute, two foreign keys named C_ID from CUSTOMERS table and E_ID from EMPLOYEES table. Lastly, ORDER_HISTORY table contains two foreign keys named T_ID from TABLES table and O_ID from ORDERS table.

Application of the Database in Real Life

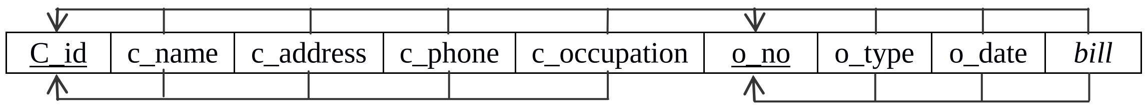
By maintaining a database, it is possible to see the bookings or orders of a restaurant and it helps to serve the orders on time to the customers. A database also helps to find out all the employees work-flow . So, it is easier to monitor employee’s activity by using a database. Most popular food items and the least popular food items can be easily determined by using a database. A database helps the a restaurant manager to analyze previous data and come to a decision easily and in a short time. This is how a database ensures efficiency and proper management of a restaurant.

Entity Relationship Diagram



Normalization

Relation: *Place*



C_id	c_name	c_address	c_phone	c_occupation	o_no	o_type	o_date	*bill*

1NF:

C_id	c_name	c_address	c_phone	c_occupation	o_no	o_type	o_date	*bill*

2NF:

Customers

C_id	c_name	c_address	c_phone	c_occupation

Orders

o_no	o_type	o_date	bill	c_id

3NF:

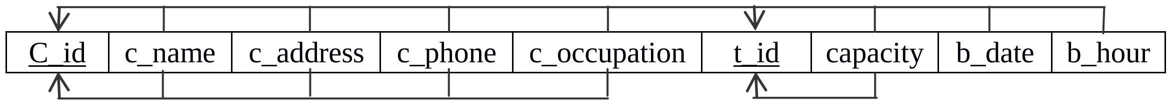
Customers

C_id	c_name	c_address	c_phone	c_occupation

Orders

o_no	o_type	o_date	bill	c_id

Relation: *Book*



C_id	c_name	c_address	c_phone	c_occupation	t_id	capacity	b_date	b_hour

1NF:

c_id	c_name	c_address	c_phone	c_occupation	t_id	capacity	b_date	b_hour

2NF:

Customers

c_id	c_name	c_address	c_phone	c_occupation

Tables

t_id	capacity

Booking

c_id	t_id	b_date	b_hour

3NF:

Customers

C_id	c_name	c_address	c_phone	c_occupation

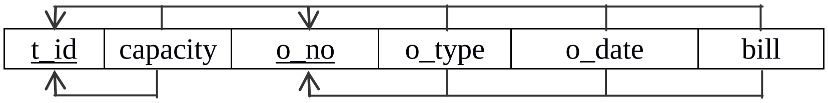
Tables

t_id	capacity

Booking

c_id	t_id	b_date	b_hour

Relation: *Have*



t_id	capacity	o_no	o_type	o_date	bill

1NF:

t_id	capacity	o_no	o_type	o_date	bill

2NF:

Tables

t_id	capacity

Orders

o_no	o_type	o_date	bill

Order_history

t_id	o_no

3NF:

Tables

t_id	capacity

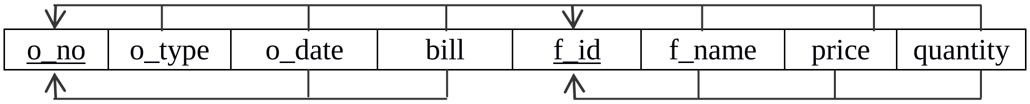
Orders

o_no	o_type	o_date	bill

Order_history

t_id	o_no

Relation: *Contain*



o_no	o_type	o_date	bill	f_id	f_name	price	quantity

1NF:

o_no	o_type	o_date	bill	f_id	f_name	price	quantity

2NF:

Orders

o_no	o_type	o_date	bill

Foods

f_id	f_name	price

Items

o_no	f_id	quantity

3NF:

Orders

o_no	o_type	o_date	bill

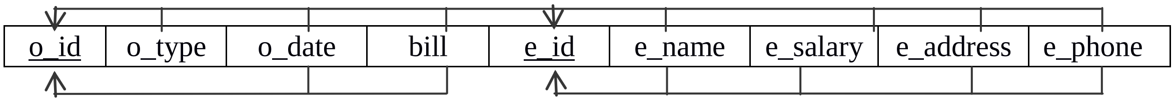
Foods

f_id	f_name	price

Items

o_no	f_id	quantity

Relation: *Taken by*



o_id	o_type	o_date	bill	e_id	e_name	e_salary	e_address	e_phone

1NF:

o_id	o_type	o_date	bill	e_id	e_name	e_salary	e_address	e_phone

2NF:

Orders

o_id	o_type	o_date	bill	e_id

Employees

e_id	e_name	e_phone	e_address	e_salary

3NF:

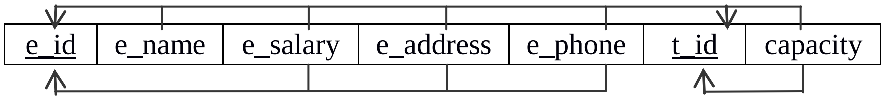
Orders

o_id	o_type	o_date	bill	e_id

Employees

e_id	e_name	e_phone	e_address	e_salary

Relation: *Maintain*



e_id	e_name	e_salary	e_address	e_phone	t_id	capacity

1NF:

e_id	e_name	e_salary	e_address	e_phone	t_id	capacity

2NF:

Tables

t_id	capacity	e_id

Employees

e_id	e_name	e_phone	e_address	e_salary

3NF:

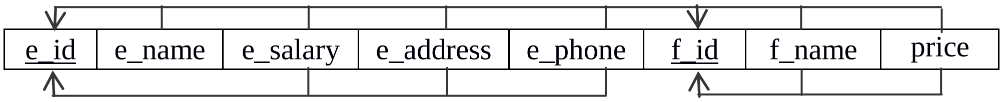
Tables

t_id	capacity	e_id

Employees

e_id	e_name	e_phone	e_address	e_salary

Relation: *Prepare*



e_id	e_name	e_salary	e_address	e_phone	f_id	f_name	price

1NF:

e_id	e_name	e_salary	e_address	e_phone	f_id	f_name	price

2NF:

Employees

e_id	e_name	e_salary	e_address	e_phone

Foods

f_id	f_name	price	e_id

3NF:

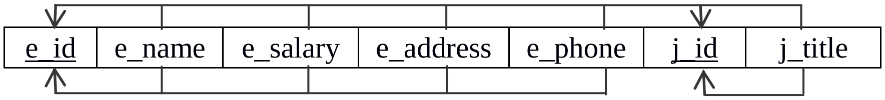
Employees

e_id	e_name	e_salary	e_address	e_phone

Foods

f_id	f_name	price	e_id

Relation: *Assign*



e_id	e_name	e_salary	e_address	e_phone	j_id	j_title

1NF:

e_id	e_name	e_salary	e_address	e_phone	j_id	j_title

2NF:

Employees

e_id	e_name	e_salary	e_address	e_phone	j_id

Jobs

j_id	j_title

3NF:

Employees

e_id	e_name	e_salary	e_address	e_phone	j_id

Jobs

j_id	j_title

Normalized Tables:

Jobs

j_id	j_title

Employees

e_id	e_name	e_phone	e_address	e_salary	j_id

Foods

f_id	f_name	price	e_id

Tables

t_id	capacity	e_id

Booking

c_id	t_id	b_date	b_hour

Customers

c_id	c_name	c_address	c_phone	c_occupation

Orders

o_id	o_type	o_date	bill	c_id	e_id

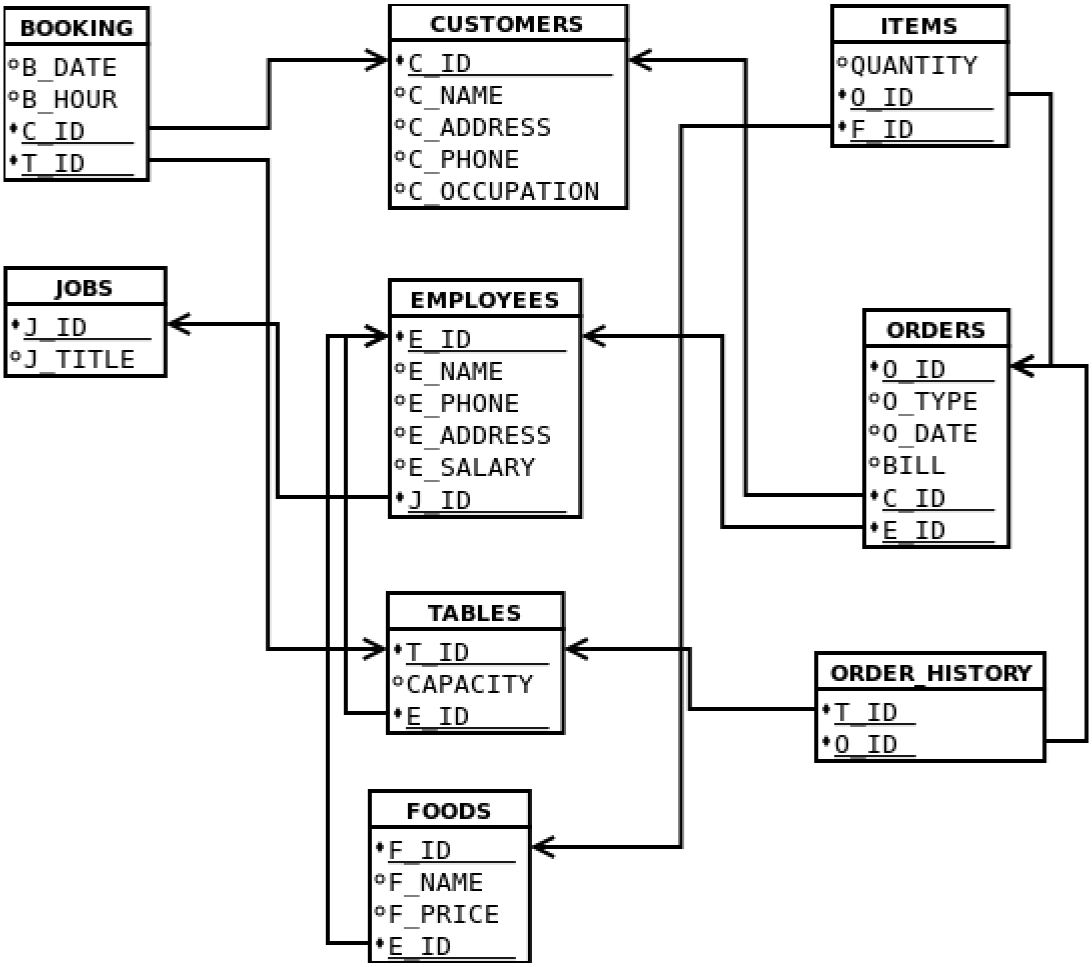
Order_history

t_id	o_id

Items

o_id	f_id	quantity

Schema


BOOKING
B_DATE
B_HOUR
C_ID
T_ID
CUSTOMERS
C_ID
C_NAME
C_ADDRESS
C_PHONE
C_OCCUPATION
ITEMS
QUANTITY
O_ID
F_ID
JOBS
J_ID
J_TITLE
EMPLOYEES
E_ID
E_NAME
E_PHONE
E_ADDRESS
E_SALARY
J_ID
ORDERS
O_ID
O_TYPE
O_DATE
BILL
C_ID
E_ID
TABLES
T_ID
CAPACITY
E_ID
ORDER_HISTORY
T_ID
O_ID
FOODS
F_ID
F_NAME
F_PRICE
E_ID

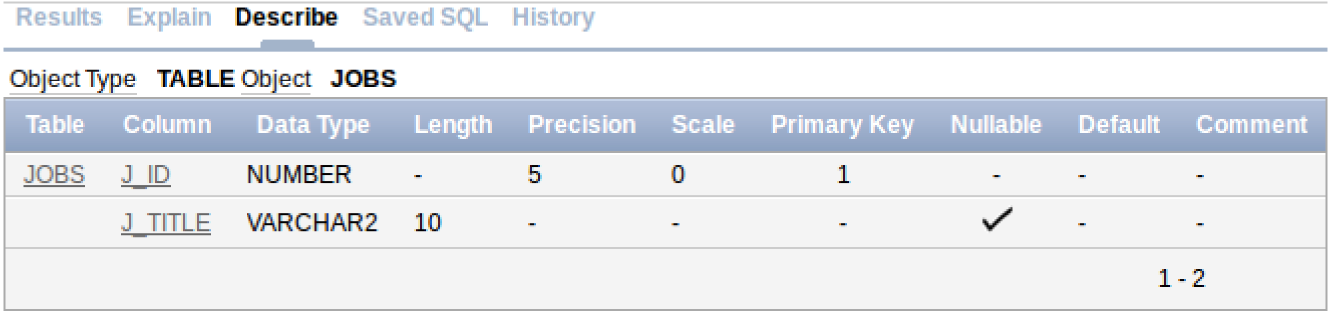
DDL

```
CREATE TABLE jobs(

        j_id NUMBER (5),

        j_title VARCHAR2 (10),

        CONSTRAINT jobs_j_id_pk PRIMARY KEY (j_id)

);


CREATE TABLE employees(

        e_id NUMBER (7),

        e_name VARCHAR2 (20) NOT NULL,

        e_phone NUMBER (11)NOT NULL,

        e_address VARCHAR2 (25) NOT NULL,

        e_salary NUMBER (6),

        j_id NUMBER (5) NOT NULL,

        CONSTRAINT employees_e_id_pk PRIMARY KEY (e_id),

        CONSTRAINT employees_j_id_fk FOREIGN KEY (j_id) REFERENCES jobs (j_id)

);


CREATE TABLE tables(

        t_id NUMBER (5),

        capacity NUMBER (2) NOT NULL,

        e_id NUMBER (7) NOT NULL,

        CONSTRAINT tables_t_id_pk PRIMARY KEY (t_id),

        CONSTRAINT tables_e_id_fk FOREIGN KEY (e_id) REFERENCES employees (e_id)

);
```

```
CREATE TABLE foods(
        f_id NUMBER (7),
        f_name VARCHAR2 (10) UNIQUE,
        f_price NUMBER (5) NOT NULL,
        e_id NUMBER (7) NOT NULL,
        CONSTRAINT foods_f_id_pk PRIMARY KEY (f_id),
        CONSTRAINT foods_e_id_fk FOREIGN KEY (e_id) REFERENCES employees (e_id)
);

CREATE TABLE customers(
        c_id NUMBER (7),
        c_name VARCHAR2 (20),
        c_phone NUMBER (11),
        c_address VARCHAR2 (25),
        c_occupation VARCHAR2 (6),
        CONSTRAINT customers_c_id_pk PRIMARY KEY (c_id)
);

CREATE TABLE orders(
        o_id NUMBER (7),
        o_type VARCHAR2 (10),
        o_date DATE DEFAULT SYSDATE,
        c_id NUMBER (7),
        e_id NUMBER (7),
        CONSTRAINT orders_o_id_pk PRIMARY KEY (o_id),
        CONSTRAINT orders_cid_fk FOREIGN KEY (c_id) REFERENCES customers (c_id),
        CONSTRAINT orders_eid_fk FOREIGN KEY (e_id) REFERENCES employees (e_id)
);
```

```
CREATE TABLE order_history(
        t_id NUMBER (5),
        o_id NUMBER (7),
        CONSTRAINT order_history_PK PRIMARY KEY (t_id, O_id),
        CONSTRAINT order_history_tid_fk FOREIGN KEY (t_id) REFERENCES tables (t_id),
        CONSTRAINT order_history_oid_fk FOREIGN KEY (o_id) REFERENCES orders (o_id)
);

CREATE TABLE items(
        quantity NUMBER (4) NOT NULL,
        o_id NUMBER (7),
        f_id NUMBER (7),
        CONSTRAINT items_oid_fid_pk PRIMARY KEY (o_id, f_id),
        CONSTRAINT items_oid_fk FOREIGN KEY (o_id) REFERENCES orders (o_id),
        CONSTRAINT items_fid_fk FOREIGN KEY (f_id) REFERENCES foods (f_id)
);

CREATE TABLE booking(
        b_date DATE,
        b_hour NUMBER(2),
        c_id NUMBER (7),
        t_id NUMBER (5),
        CONSTRAINT booking_cid_tid_pk PRIMARY KEY (c_id, t_id),
        CONSTRAINT booing_cid_fk FOREIGN KEY (c_id) REFERENCES customers(c_id),
        CONSTRAINT booing_tid_fk FOREIGN KEY (t_id) REFERENCES tables (t_id)
);
```

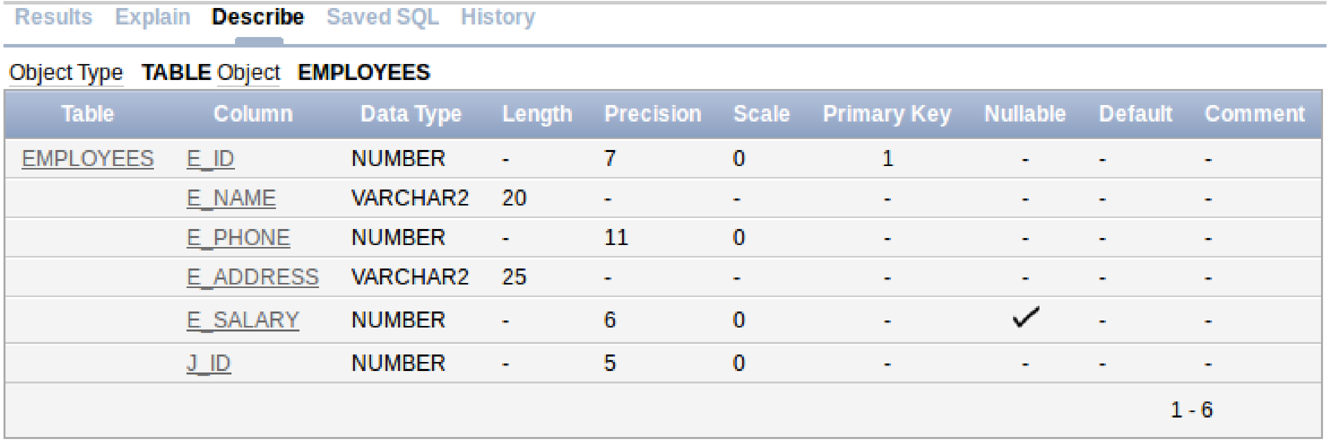
Table Descriptions



Object Type **TABLE** Object **JOBS**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
JOBS	J_ID	NUMBER	-	5	0	1	-	-	-
	J_TITLE	VARCHAR2	10	-	-	-	✓	-	-

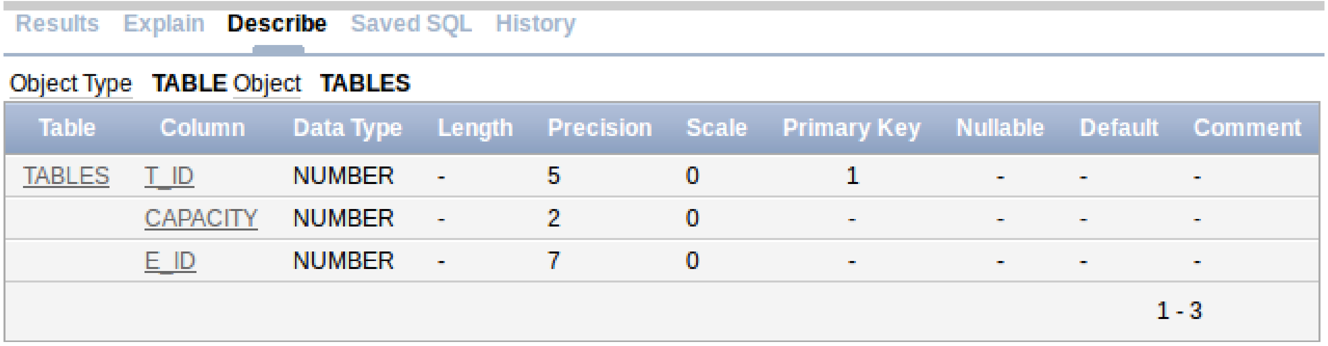
1 - 2



Object Type **TABLE** Object **EMPLOYEES**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EMPLOYEES	E_ID	NUMBER	-	7	0	1	-	-	-
	E_NAME	VARCHAR2	20	-	-	-	-	-	-
	E_PHONE	NUMBER	-	11	0	-	-	-	-
	E_ADDRESS	VARCHAR2	25	-	-	-	-	-	-
	E_SALARY	NUMBER	-	6	0	-	✓	-	-
	J_ID	NUMBER	-	5	0	-	-	-	-

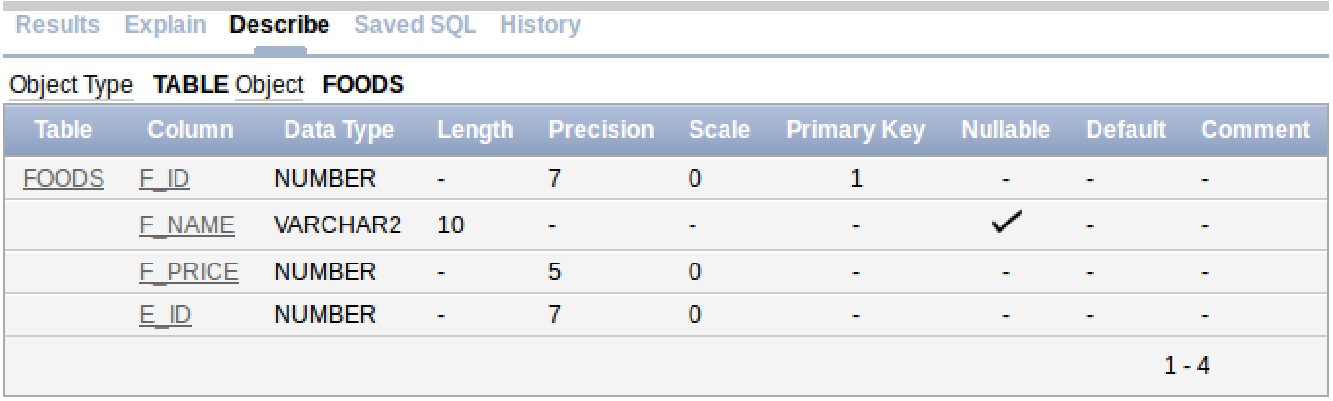
1 - 6



Object Type **TABLE** Object **TABLES**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TABLES	T_ID	NUMBER	-	5	0	1	-	-	-
	CAPACITY	NUMBER	-	2	0	-	-	-	-
	E_ID	NUMBER	-	7	0	-	-	-	-

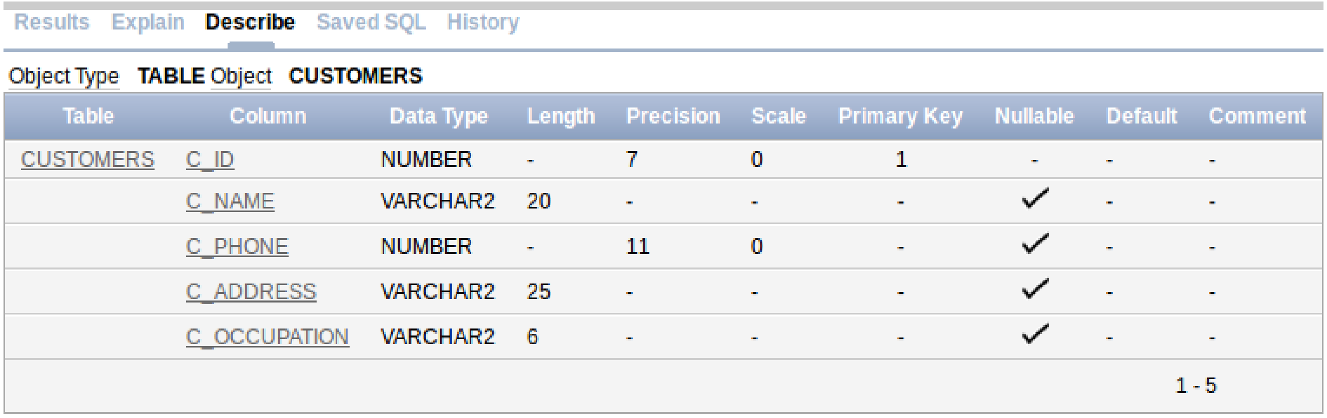
1 - 3



Object Type **TABLE** Object **FOODS**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
FOODS	F_ID	NUMBER	-	7	0	1	-	-	-
	F_NAME	VARCHAR2	10	-	-	-	✓	-	-
	F_PRICE	NUMBER	-	5	0	-	-	-	-
	E_ID	NUMBER	-	7	0	-	-	-	-

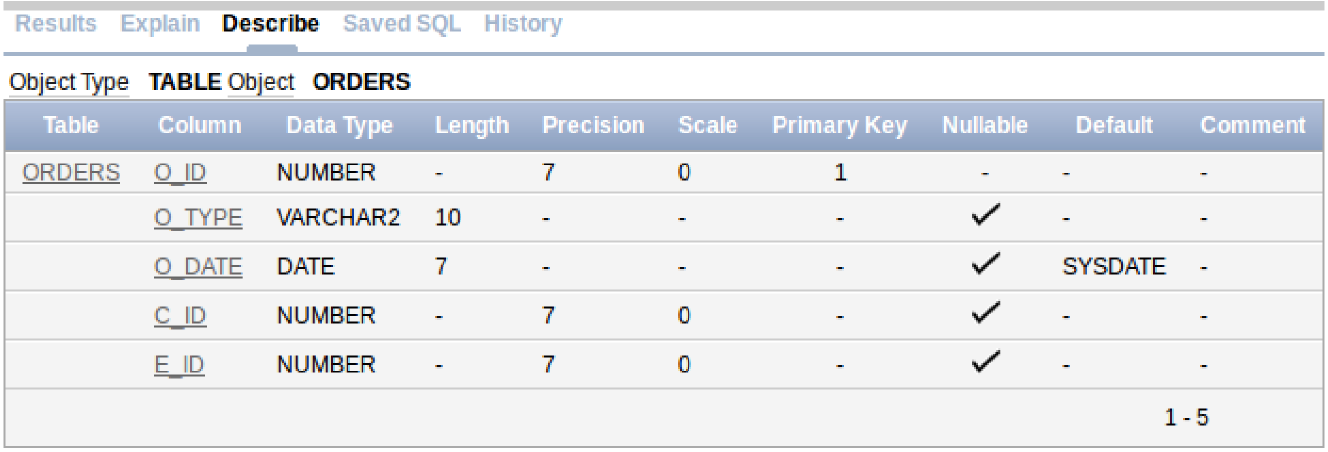
1 - 4



Object Type **TABLE** Object **CUSTOMERS**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMERS	C_ID	NUMBER	-	7	0	1	-	-	-
	C_NAME	VARCHAR2	20	-	-	-	✓	-	-
	C_PHONE	NUMBER	-	11	0	-	✓	-	-
	C_ADDRESS	VARCHAR2	25	-	-	-	✓	-	-
	C_OCCUPATION	VARCHAR2	6	-	-	-	✓	-	-

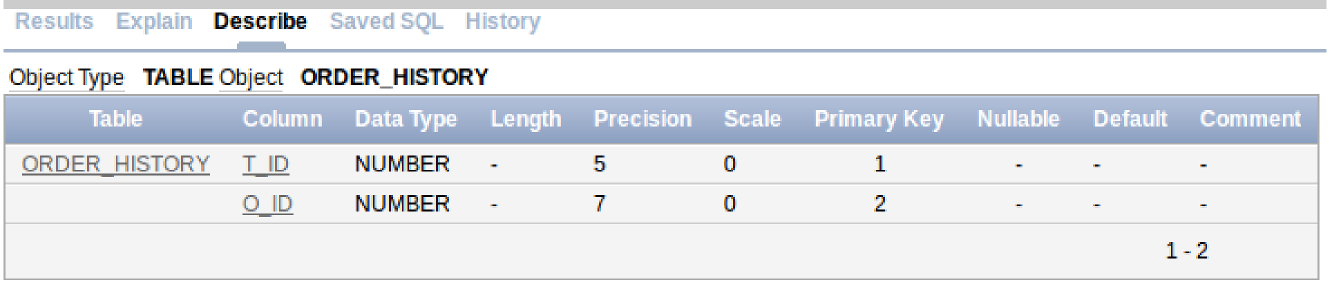
1 - 5



Object Type **TABLE** Object **ORDERS**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ORDERS	O_ID	NUMBER	-	7	0	1	-	-	-
	O_TYPE	VARCHAR2	10	-	-	-	✓	-	-
	O_DATE	DATE	7	-	-	-	✓	SYSDATE	-
	C_ID	NUMBER	-	7	0	-	✓	-	-
	E_ID	NUMBER	-	7	0	-	✓	-	-

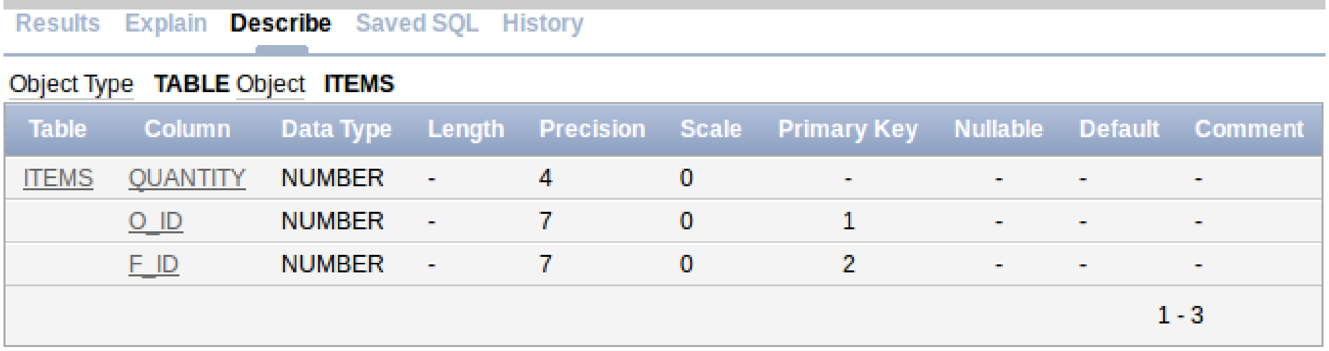
1 - 5



Object Type **TABLE** Object **ORDER_HISTORY**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ORDER_HISTORY	T_ID	NUMBER	-	5	0	1	-	-	-
	O_ID	NUMBER	-	7	0	2	-	-	-

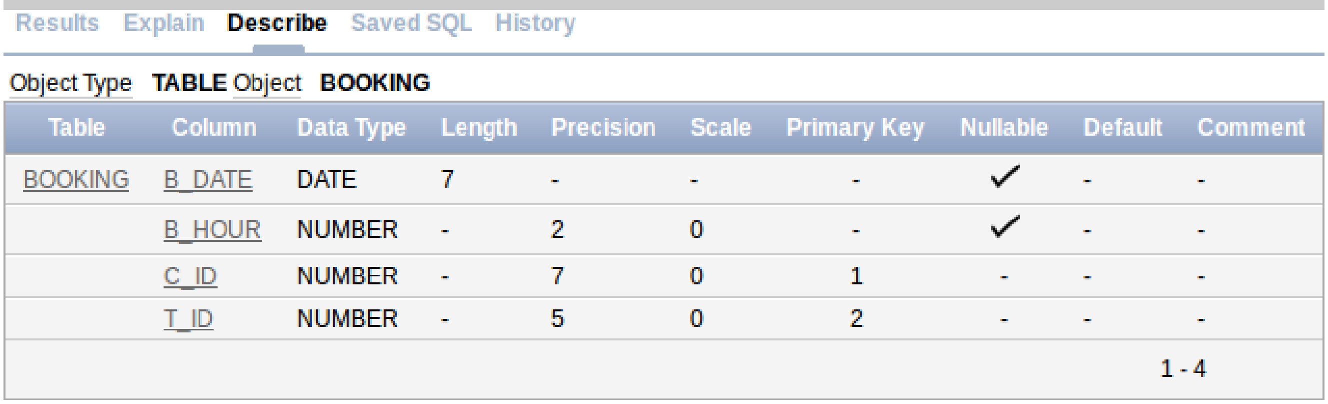
1 - 2



Object Type **TABLE** Object **ITEMS**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ITEMS	QUANTITY	NUMBER	-	4	0	-	-	-	-
	O_ID	NUMBER	-	7	0	1	-	-	-
	F_ID	NUMBER	-	7	0	2	-	-	-

1 - 3



Object Type **TABLE** Object **BOOKING**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BOOKING	B_DATE	DATE	7	-	-	-	✓	-	-
	B_HOUR	NUMBER	-	2	0	-	✓	-	-
	C_ID	NUMBER	-	7	0	1	-	-	-
	T_ID	NUMBER	-	5	0	2	-	-	-

1 - 4

Value Insertion

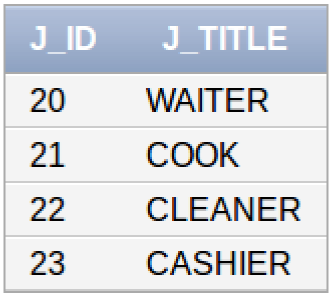
```
INSERT INTO jobs (j_id, j_title) VALUES (20, 'WAITER');

INSERT INTO jobs (j_id, j_title) VALUES (21, 'COOK');

INSERT INTO jobs (j_id, j_title) VALUES (22, 'CLEANER');

INSERT INTO jobs (j_id, j_title) VALUES (23, 'CASHIER');
```

```
INSERT INTO customers (c_id, c_name, c_address, c_phone, c_occupation) VALUES (101, 'Tanzirul Haque', 'Basundhara, Dhaka', 01788284381, 'STU');

INSERT INTO customers (c_id, c_name, c_address, c_phone, c_occupation) VALUES (102, 'Sabbir Hosssain', 'Mohakhali, Dhaka', 01788284381, 'STU');

INSERT INTO customers (c_id, c_name, c_address, c_phone, c_occupation) VALUES (103, 'Sakib Hosssain', 'Baridhara, Dhaka', 01521301871, 'STU');

INSERT INTO customers (c_id, c_name, c_address, c_phone, c_occupation) VALUES (104, 'Al Amin', 'Notun Bazar, Dhaka', 01521318596, 'STU');

INSERT INTO customers (c_id, c_name, c_address, c_phone, c_occupation) VALUES (105, 'Muktadir Udoy', 'Basundhara, Dhaka', 01773709191, 'DOC');

INSERT INTO customers (c_id, c_name, c_address, c_phone, c_occupation) VALUES (106, 'Jason James', 'Basundhara, Dhaka', 01554369526, 'ENG');
```

```
INSERT INTO employees (e_id, e_name, e_phone, e_address, e_salary, j_id) VALUES (1, 'Abdal Mumit', 01788523695, 'Nikunjo, Dhaka', 15000, 20);

INSERT INTO employees (e_id, e_name, e_phone, e_address, e_salary, j_id) VALUES (2, 'Ar Rafi', 01756295398, 'Nikunjo, Dhaka', 19000, 21);

INSERT INTO employees (e_id, e_name, e_phone, e_address, e_salary, j_id) VALUES (3, 'Fokhrul Hridoy', 01653269852, 'Uttara, Dhaka', 9000, 22);

INSERT INTO employees (e_id, e_name, e_phone, e_address, e_salary, j_id) VALUES (4, 'Obida Alauddin', 01856296541, 'Nikunjo, Dhaka', 21000, 23);
INSERT INTO foods (f_id, f_name, f_price, e_id) VALUES (1, 'BURGER', 450, 2);
INSERT INTO foods (f_id, f_name, f_price, e_id) VALUES (2, 'SANDWICH', 180, 2);
INSERT INTO foods (f_id, f_name, f_price, e_id) VALUES (3, 'PIZZA', 880, 2);
INSERT INTO foods (f_id, f_name, f_price, e_id) VALUES (4, 'PASTA', 300, 2);


INSERT INTO tables (t_id, capacity, e_id) VALUES (101, 4, 1);
INSERT INTO tables (t_id, capacity, e_id) VALUES (102, 4, 1);
INSERT INTO tables (t_id, capacity, e_id) VALUES (103, 2, 1);
INSERT INTO tables (t_id, capacity, e_id) VALUES (104, 10, 1);


INSERT INTO booking (b_date, b_hour, c_id, t_id) VALUES ('12-12-2016', 2, 101, 101);
INSERT INTO booking (b_date, b_hour, c_id, t_id) VALUES ('12-13-2016', 1, 103, 102);
INSERT INTO booking (b_date, b_hour, c_id, t_id) VALUES ('12-14-2016', 6, 104, 104);


INSERT INTO orders (o_id, o_type, o_date, c_id, e_id) VALUES (1, 'IN HOUSE', '12-12-2016', 101, 1);
INSERT INTO orders (o_id, o_type, o_date, c_id, e_id) VALUES (2, 'PARCEL', '12-12-2016', 102, 1);
INSERT INTO orders (o_id, o_type, o_date, c_id, e_id) VALUES (3, 'IN HOUSE', '12-13-2016', 103, 1);
INSERT INTO orders (o_id, o_type, o_date, c_id, e_id) VALUES (5, 'IN HOUSE', '12-14-2016', 104, 1);
INSERT INTO orders (o_id, o_type, o_date, c_id, e_id) VALUES (4, 'PARCEL', '12-14-2016', 105, 1);
INSERT INTO orders (o_id, o_type, o_date, c_id, e_id) VALUES (6, 'PARCEL', '12-14-2016', 106, 1);
INSERT INTO items (o_id, f_id, quantity) VALUES (1, 3, 4);
INSERT INTO items (o_id, f_id, quantity) VALUES (2, 3, 2);
INSERT INTO items (o_id, f_id, quantity) VALUES (3, 4, 3);
INSERT INTO items (o_id, f_id, quantity) VALUES (4, 3, 4);
INSERT INTO items (o_id, f_id, quantity) VALUES (5, 3, 10);
INSERT INTO items (o_id, f_id, quantity) VALUES (5, 1, 10);
INSERT INTO items (o_id, f_id, quantity) VALUES (6, 2, 1);


INSERT INTO order_history (t_id, o_id) VALUES (101, 1);
INSERT INTO order_history (t_id, o_id) VALUES (102, 3);
INSERT INTO order_history (t_id, o_id) VALUES (104, 5);
```

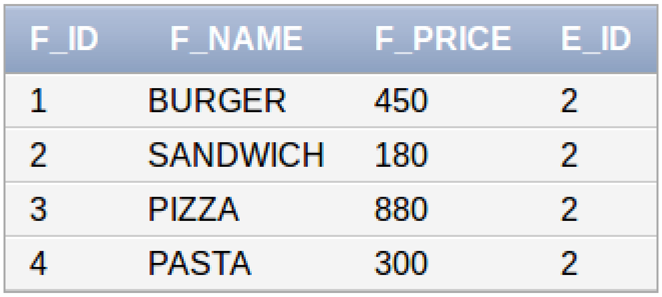
Tables with Data

JOBS table:



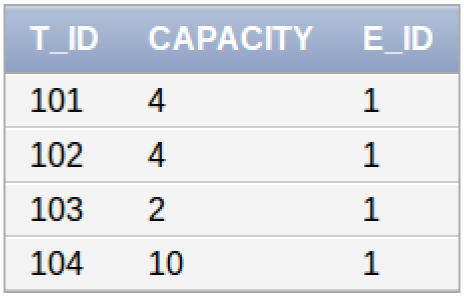
J_ID	J_TITLE
20	WAITER
21	COOK
22	CLEANER
23	CASHIER

FOODS table:



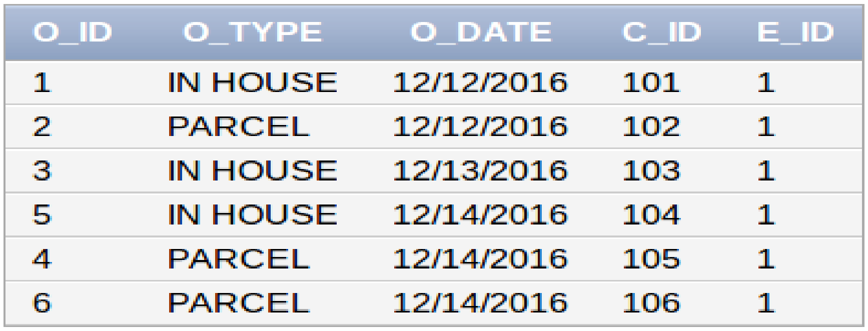
F_ID	F_NAME	F_PRICE	E_ID
1	BURGER	450	2
2	SANDWICH	180	2
3	PIZZA	880	2
4	PASTA	300	2

TABLES table:



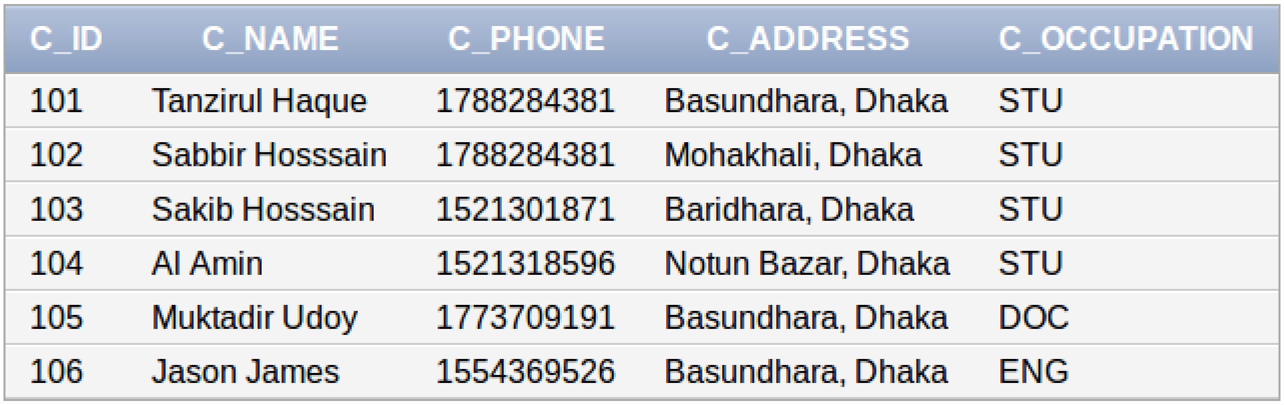
T_ID	CAPACITY	E_ID
101	4	1
102	4	1
103	2	1
104	10	1

ORDERS table:



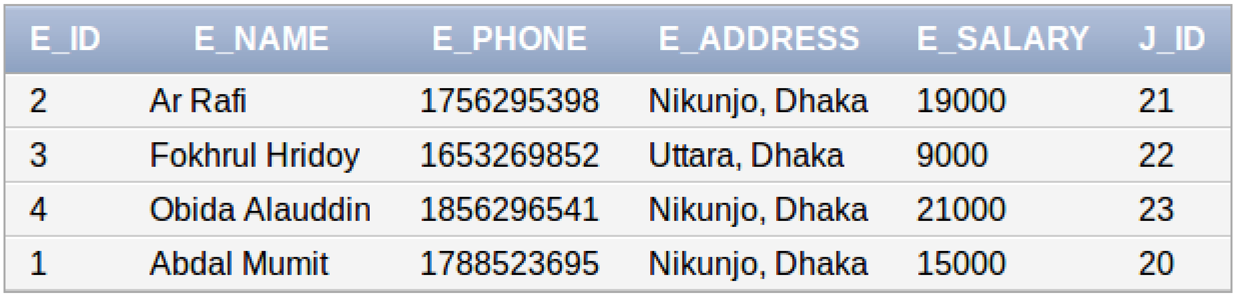
O_ID	O_TYPE	O_DATE	C_ID	E_ID
1	IN HOUSE	12/12/2016	101	1
2	PARCEL	12/12/2016	102	1
3	IN HOUSE	12/13/2016	103	1
5	IN HOUSE	12/14/2016	104	1
4	PARCEL	12/14/2016	105	1
6	PARCEL	12/14/2016	106	1

CUSTOMERS table:



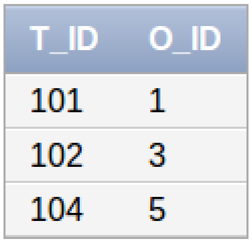
C_ID	C_NAME	C_PHONE	C_ADDRESS	C_OCCUPATION
101	Tanzirul Haque	1788284381	Basundhara, Dhaka	STU
102	Sabbir Hosssain	1788284381	Mohakhali, Dhaka	STU
103	Sakib Hosssain	1521301871	Baridhara, Dhaka	STU
104	Al Amin	1521318596	Notun Bazar, Dhaka	STU
105	Muktadir Udoy	1773709191	Basundhara, Dhaka	DOC
106	Jason James	1554369526	Basundhara, Dhaka	ENG

EMPLOYEES table:



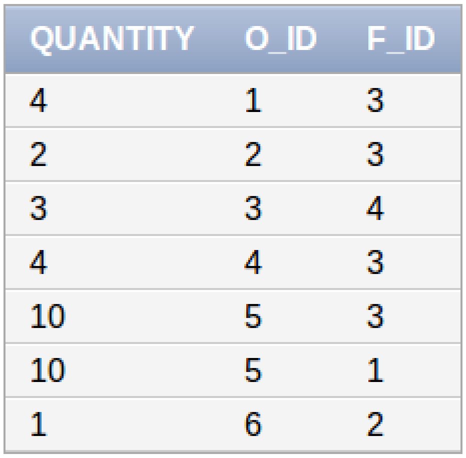
E_ID	E_NAME	E_PHONE	E_ADDRESS	E_SALARY	J_ID
2	Ar Rafi	1756295398	Nikunjo, Dhaka	19000	21
3	Fokhrul Hridoy	1653269852	Uttara, Dhaka	9000	22
4	Obida Alauddin	1856296541	Nikunjo, Dhaka	21000	23
1	Abdal Mumit	1788523695	Nikunjo, Dhaka	15000	20

ORDER_HISTORY table:



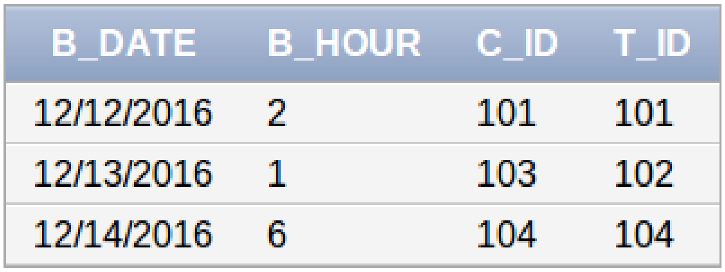
T_ID	O_ID
101	1
102	3
104	5

ITEMS table:



QUANTITY	O_ID	F_ID
4	1	3
2	2	3
3	3	4
4	4	3
10	5	3
10	5	1
1	6	2

BOOKING table:

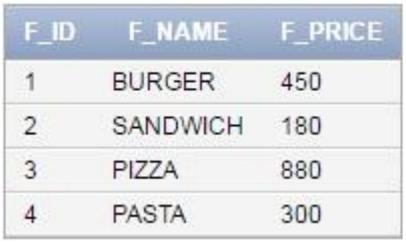


B_DATE	B_HOUR	C_ID	T_ID
12/12/2016	2	101	101
12/13/2016	1	103	102
12/14/2016	6	104	104

SQL Queries

1.Show menu of the restaurant.

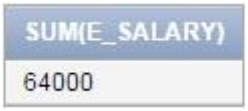
SELECT f_id, f_name, f_price FROM foods;



F_ID	F_NAME	F_PRICE
1	BURGER	450
2	SANDWICH	180
3	PIZZA	880
4	PASTA	300

2.Show total salary of all employees.

SELECT SUM(e_salary) FROM employees;



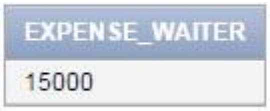
SUM(E_SALARY)
64000

3.Show the expense of the restaurant in waiter purpose.

SELECT SUM(e_salary) Expense_Waiter FROM employees

WHERE

j_id = (SELECT j_id FROM jobs WHERE j_title = 'Waiter');



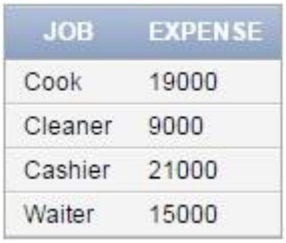
EXPENSE_WAITER
15000

4.Show the expenses of the restaurant behind employees , job wise.

SELECT j_title Job, SUM(e_salary) Expense

FROM

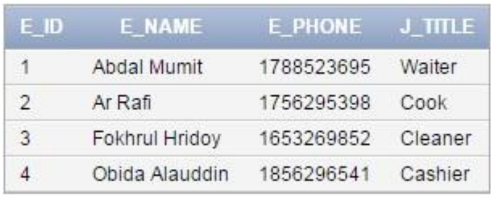
employees NATURAL JOIN jobs GROUP BY j_title;



JOB	EXPENSE
Cook	19000
Cleaner	9000
Cashier	21000
Waiter	15000

5.Show employee detail (id, name, phone, job_title).

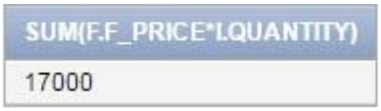
```
SELECT e.e_id, e.e_name, e.e_phone, j.j_title
FROM
employees e, jobs j WHERE e.j_id = j.j_id;
```



E_ID	E_NAME	E_PHONE	J_TITLE
1	Abdal Mumit	1788523695	Waiter
2	Ar Rafi	1756295398	Cook
3	Fokhrul Hridoy	1653269852	Cleaner
4	Obida Alauddin	1856296541	Cashier

6.Show the total sell in 14 Dec/16 (Without help of View).

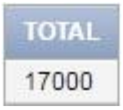
```
SELECT SUM (f.f_price * i.quantity) FROM foods f, items i
WHERE
   i.f_id = f.f_id
               AND
i.o_id IN (SELECT o_id FROM orders o WHERE o_date = '12/14/2016');
```



SUM(F.F_PRICE*I.QUANTITY)
17000

7.Show the total sell in 14 Dec/16 (With the help of View).

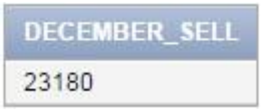
```
SELECT total FROM sell_detail WHERE s_date = '12/14/2016';
```



TOTAL
17000

8.Show the total sell in december.

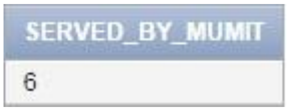
```
SELECT SUM(total) AS DECEMBER_SELL FROM sell_detail
WHERE s_date BETWEEN '12/01/2016' AND '12/31/2016';
```



DECEMBER_SELL
23180

9.Show the total number of customers served by Abdal Mumit.

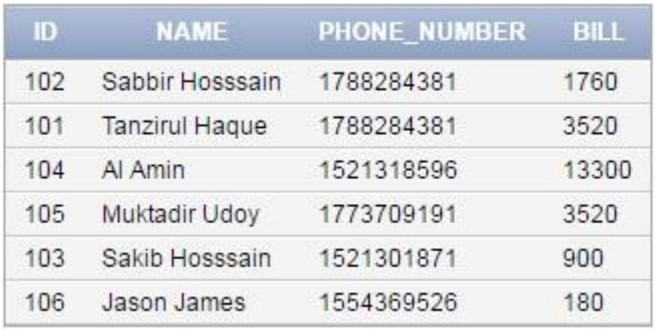
```
SELECT COUNT(o_id) Served_by_Mumit FROM orders
WHERE
e_id = (SELECT e_id FROM employees WHERE e_name = 'Abdal Mumit');
```



SERVED_BY_MUMIT
6

10.Show Customer id, name, phone number and bill for all customers.

```
SELECT bd.id AS ID, c.c_name AS NAME, c.c_phone AS PHONE_NUMBER, bd.bill AS BILL
FROM
bill_detail bd, customers c WHERE c.c_id = bd.id;
```



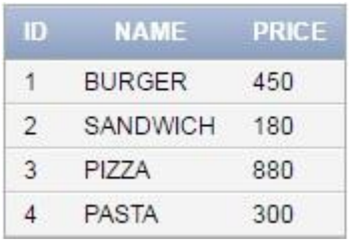
ID	NAME	PHONE_NUMBER	BILL
102	Sabbir Hosssain	1788284381	1760
101	Tanzirul Haque	1788284381	3520
104	Al Amin	1521318596	13300
105	Muktadir Udoy	1773709191	3520
103	Sakib Hosssain	1521301871	900
106	Jason James	1554369526	180

Views

Simple:

1.

```
CREATE VIEW   RES_MENU
        (ID, NAME, PRICE) AS
SELECT f_id, f_name, f_price FROM foods;
```



ID	NAME	PRICE
1	BURGER	450
2	SANDWICH	180
3	PIZZA	880
4	PASTA	300

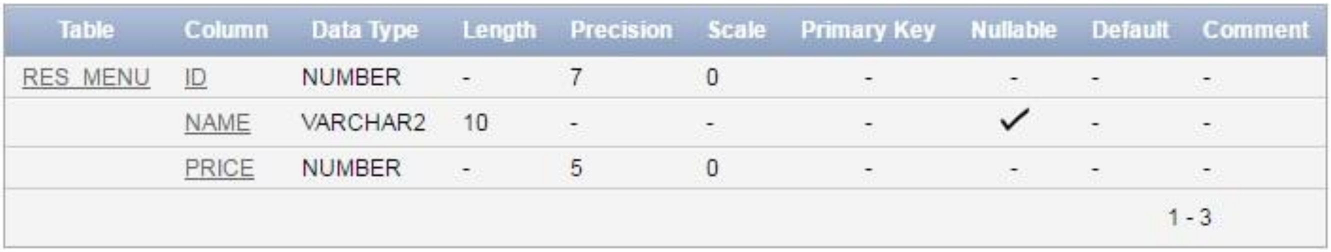
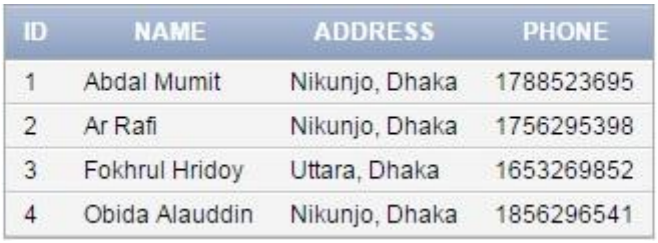


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
RES_MENU	ID	NUMBER	-	7	0	-	-	-	-
	NAME	VARCHAR2	10	-	-	-	✓	-	-
	PRICE	NUMBER	-	5	0	-	-	-	-
									1 - 3

2.

```
CREATE VIEW   EMP_LIST
          (ID, NAME, ADDRESS, PHONE)  AS
SELECT e_id, e_name, e_address, e_phone FROM employees;
```



ID	NAME	ADDRESS	PHONE
1	Abdal Mumit	Nikunjo, Dhaka	1788523695
2	Ar Rafi	Nikunjo, Dhaka	1756295398
3	Fokhrul Hridoy	Uttara, Dhaka	1653269852
4	Obida Alauddin	Nikunjo, Dhaka	1856296541

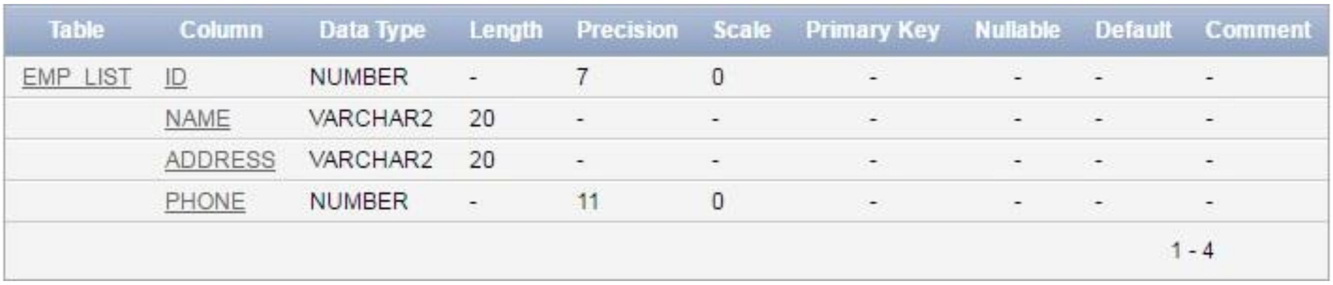
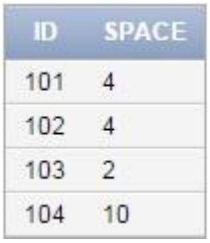


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EMP_LIST	ID	NUMBER	-	7	0	-	-	-	-
	NAME	VARCHAR2	20	-	-	-	-	-	-
	ADDRESS	VARCHAR2	20	-	-	-	-	-	-
	PHONE	NUMBER	-	11	0	-	-	-	-
									1 - 4

3.

```
CREATE VIEW   TABLE_DETAIL
                        (ID, SPACE)  AS
SELECT t_id, capacity FROM tables;
```



ID	SPACE
101	4
102	4
103	2
104	10

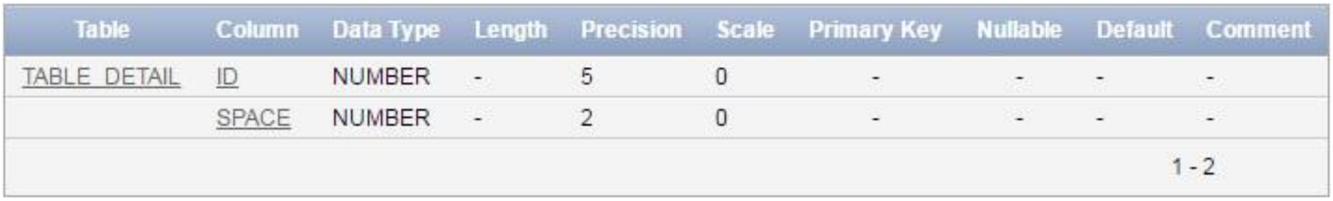


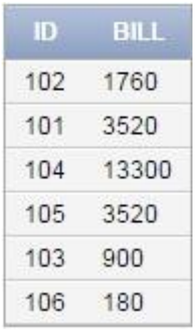
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TABLE_DETAIL	ID	NUMBER	-	5	0	-	-	-	-
	SPACE	NUMBER	-	2	0	-	-	-	-

1 - 2

Complex:

4.

```
CREATE VIEW   bill_detail
                  (ID, BILL) AS
SELECT c.c_id, SUM (f.f_price * i.quantity) FROM
                    customers c, foods f, items i, orders o
      WHERE c.c_id  =  o.c_id AND o.o_id =  i.o_id
      AND   i.f_id  =  f.f_id      GROUP BY   c.c_id;
```



ID	BILL
102	1760
101	3520
104	13300
105	3520
103	900
106	180

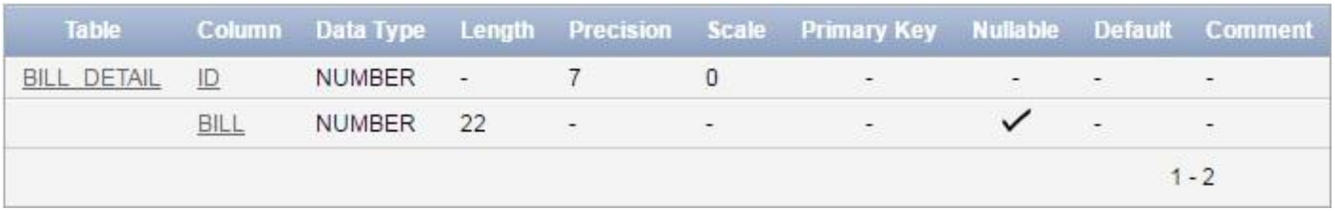
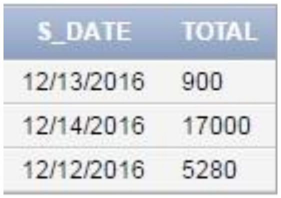


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BILL_DETAIL	ID	NUMBER	-	7	0	-	-	-	-
	BILL	NUMBER	22	-	-	-	✓	-	-

1 - 2

5.

```
CREATE VIEW sell_detail
    (S_DATE, TOTAL)  AS
SELECT o.o_date, SUM (f.f_price * i.quantity)
                  FROM  orders o, foods f, items i
WHERE o.o_id = i.o_id  AND  i.f_id = f.f_id
GROUP BY o.o_date;
```



S_DATE	TOTAL
12/13/2016	900
12/14/2016	17000
12/12/2016	5280

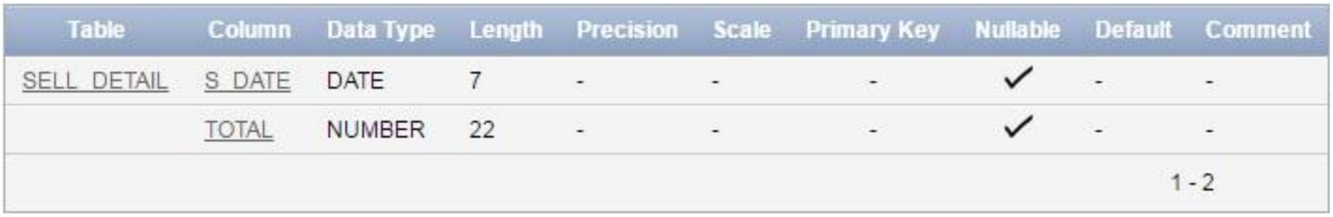


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
SELL_DETAIL	S_DATE	DATE	7	-	-	-	✓	-	-
	TOTAL	NUMBER	22	-	-	-	✓	-	-

1 - 2

Discussion

We had to face different kind of difficulties while doing the project. We got confused so many times while drawing the ERD. Finally, we could draw the ERD by studying from different sources . We also faced problems in the normalization step. An extra table was created for each many-to-many relation in the normalization process. After all of these, the database for restaurant management system was developed and still now we have confusion about derived attribute bill in the orders table. We have experienced a lot of difficulties while doing this project and we have also learned how to solve those complications. In the end, it is obvious to say that we have to improve the ordering system in future so that waiters can serve foods to customers quicker than ever and make the customers satisfied.