Mini-Project Report

PHARMACY DATABASE

By

SAMUEL ROBERT - 26
AAKASH SARDAL - 31

TE EXTC B

Course: DBMS

AY: 2020-21

Table of Contents

I. Storyline

1. This pharmacy database is based one location or pharmacy shop with medical drugs.

2. The pharmacy has inventory which contains drugs and units:
   2.1 The inventory consists of drug id, drug name, units and per unit price.
   2.2 There can be multiple units of a drug.
   2.3 Inventory keeps track of stock of medicine.

3. The pharmacy database keeps a track of the employees and stores their information in the database.
   3.1 The name, employee id is stored in the database.
   3.2 Each employee has a unique id.
   3.3 The database must keep track of the employee sales to the customer.

4. The pharmacy needs to keep a record of its customers:
   4.1 Every customer has an ID, name, phone number, address.
   4.2 The database keeps track of purchases made by the customer.
   4.3 It keeps track of the purchase date, and amount of purchase.
   4.4 The payment details are also noted with order id and item id.

5. The database must keep track of the supplier who supplies drugs to the pharmacy:
   5.1 Each supplier has name, ID, address.
   5.2 Every supplier has a unique ID.
   5.3 The database must keep track of status of the availability of drug in inventory which has been delivered by the suppliers.

6. The database keeps the track of orders:
   6.1 The orders consist of order id, order date, total unit and total price.

II. Components of Database Design

To effectively design a database, we need to have a clear understanding of the entity sets and the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties.

In this mini project, a database management system for a pharmacy is being made. The entities and their respective attributes required are as follows:

1) **Customer** – The Customer are the people who come to the pharmacy to buy medicine. The pharmacy needs to store customer ID, Name, contact number and address.
   Attributes:

   - ❖ Customer ID (int) e.g. 56121
   - ❖ Name (text) e.g. AJAY
   - ❖ Contact number (int) e.g. 9989998989
   - ❖ Customer address (text) e.g. vasai

2) **Employee**– The employee sells items from the pharmacy to the customer. The employee also manages the database.
   Attributes:

   - ❖ Employee ID (int) e.g. 104
   - ❖ Employee Name (text) e.g. AAKASH

3) **Item** – The item is found in the inventory of the pharmacy. Each and every item has its own unique id supplied by the seller
   Attributes:

   - ❖ Item ID (Int).e.g.1
   - ❖ Expiry (text) e.g. 25-03-2022
   - ❖ Status (text) e.g. delivered

4) **Inventory**– The inventory of the pharmacy contains drug details with per unit price.

Attributes:

- ❖ Drug ID (int) e.g.345
- ❖ Drug name (text) e.g. paracetamol
- ❖ unit (int) e.g. 100
- ❖ Unit price (int) e.g. 175

5) Orders – Orders keeps track of date of purchase of total amounts of unit with total price. It helps to easily identify customer's order status with order id.

Attributes:

- Order ID (int) e.g. 101
- Order date (text) e.g. 27-04-21
- Total_unit (int) e.g. 3
- Total_price(int) e.g. 350

6) Supplier – The Supplier supplies the drugs to the pharmacy with is stored in inventory with all the details of items.

Attributes:

- Supplier ID (int) e.g. 29790
- S_Name (text) e.g. cipla
- Supplier address (text) e.g. goregoan

Now that we are done describing all the entities in detail, we need to examine the **relationships** between each entity. But before we can examine the relationships, we need to be aware of a concept called cardinality.

A **cardinality** notation defines the attributes of the relationship between the entities. Cardinalities can denote that an entity is optional (for example, a sales rep could have no customers or could have many) or mandatory (for example, there must be at least one product listed in an order.)

The three main cardinalities are:

A one-to-one relationship (1:1). For example, if each customer in a database is associated with one mailing address.

A one-to-many relationship (1:M). For example, a single customer might place an order for multiple products. The customer is associated with multiple entities, but all those entities have a single connection back to the same customer.

A many-to-many relationship (M:N). For example, at a company where all call center agents work with multiple customers, each agent is associated with multiple customers, and multiple customers might also be associated with multiple agents.

Relationships and Cardinality

1) The entities **Supplier** and **Item** are connected by a relation called **Supplies**, since Supplier supplies item. Every supplier can supply many different items . An item can be supplied by many supplier. Hence it is a **one to many relationship**. Hence there is **total participation** on both sides.

2) The entities **Item** and **Orders** are connected by a relation called **have** , since Items have Orders. Every item can have several orders . Hence it is a **many to one relationship**. Each order can have several items in it. Hence there is **partial participation** in order and total on item

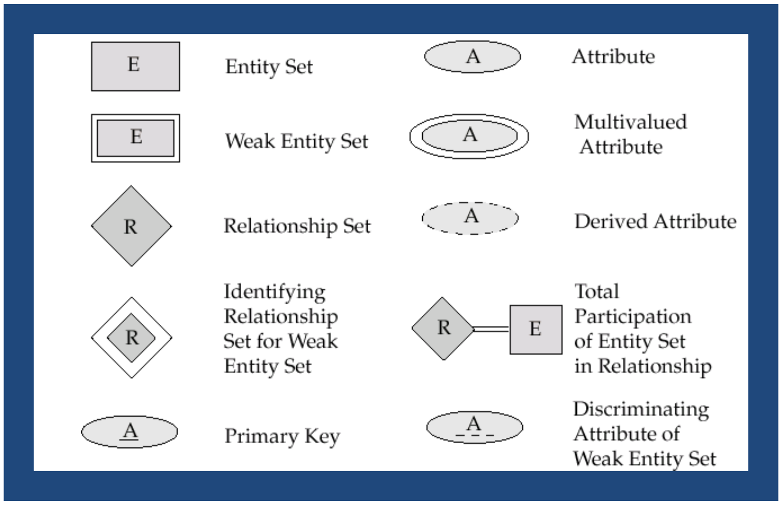
3) The entities **Item** and **Inventory** are connected by a relation called **in**, since Items are in the Inventory. There can be multiple different items in an inventory . Hence it is a **many to one relationship**. An Inventory can exist even if there are no items in it . Hence there is **partial participation** on one side (item) and **total participation** on many side (inventory).

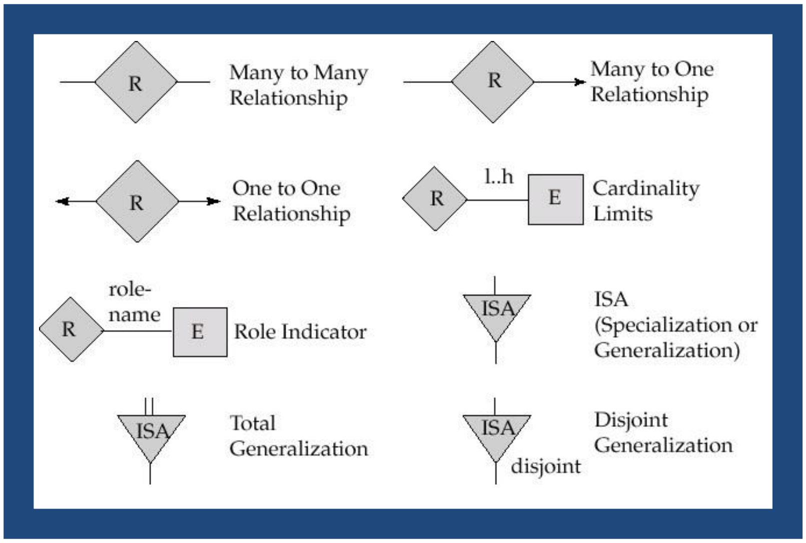
4) The entities **Orders** and **Employee** are connected by a relation called **taken_by**, since orders are taken by the employees. An order can be taken by one employee at a time but an employee can take multiple orders from the customer . Hence it is a **many to one relationship**. An employee can exist even if there is no order but an order cannot exist if it there is no employee to take the order. Hence there is **total participation** on one side (Employee) and there is **partial participation** on order.

5) The entities **Order** and **Customer** are connected by a relation called **given_by** , since orders are given by customers. Every order has a customer but one customer can place multiple orders . Hence it is a **many to one relationship**. An order must belong to a customer and a customer also must have an order. Hence there is **total participation** on both sides.

III. Entity Relationship Diagram

Entity relationship diagrams provide a visual starting point for database design that can also be used to help determine information system requirements throughout an organization. Before starting the creation of ER Diagram, we should know these conventional symbols used in it.


E
Entity Set
A
Attribute
E
Weak Entity Set
A
Multivalued Attribute
R
Relationship Set
A
Derived Attribute
R
Identifying Relationship Set for Weak Entity Set
R
E
Total Participation of Entity Set in Relationship
A
Primary Key
A
Discriminating Attribute of Weak Entity Set



Organizing the ERD in a logical way is incredibly important to increase comprehension. The main purpose of entity-relationship diagrams is to model a complex database, so learning how to create simple, logical ERDs is the key.

Entity-relationship diagrams are incredibly useful, and the following simple steps are needed to create an ER diagram:

- **Determine the entities** – The entities for this ER Diagram are Customer, Supplier, Inventory, Item, Orders, Employee

Next, we need to determine the primary key for each entity

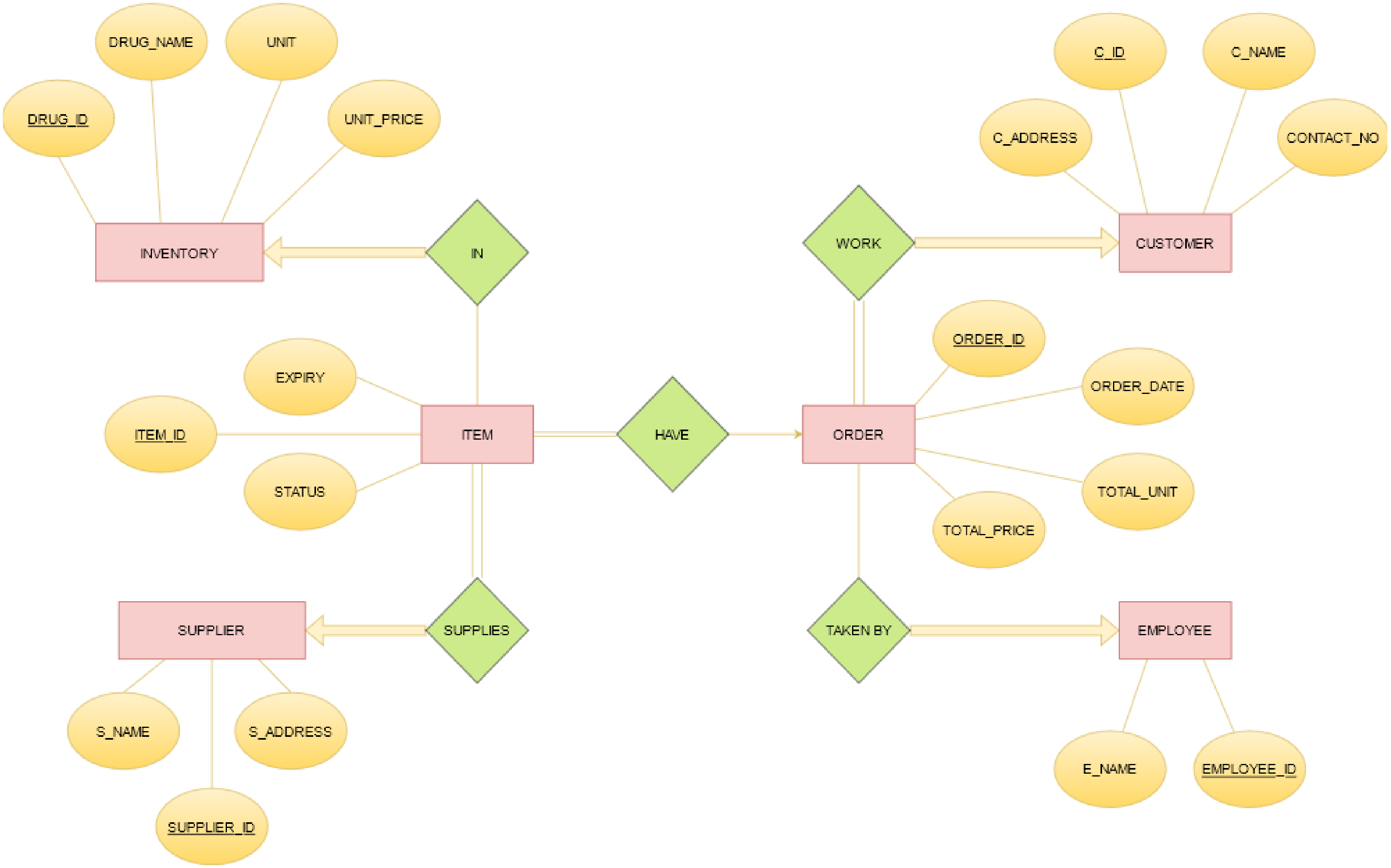
The primary key for Inventory is Drug_ID, for Employee is Emp_ID, for item is Item_ID, for Customer is Cust_ID , for Orders is Order_ID, for Supplier is Supplier_ID

Identify the relationships: Relationships highlight how entities interact with each other.

- Supplier supplies item : supplier with item : 1:N , both total
- Item have orders : item with orders : M:1, item total, order partial
- Item in inventory : Item with inventory : M:1, item partial, inventory total
- Order taken_by employee : Order with employee : M:1 order partial, employee total
- Order given_by customer : order with customer : M:1, both total

Add attributes: Attributes show specific characteristics of an entity, refining what information is important to the model.

- Customer (C_name, C_address, contact_no, CID)
- Supplier (S_name, supplier_ID, S_address)
- Inventory(Drug_id, Drug_name, unit , unit_price)
- Item (item_id, expiry, status)
- Orders (order_id, order_date, total_unit, total_price)
- Employee(Employee_name, Employee_ID)


DRUG_NAME
UNIT
DRUG_ID
UNIT_PRICE
INVENTORY
IN
C_ID
C_NAME
C_ADDRESS
CONTACT_NO
WORK
CUSTOMER
EXPIRY
ITEM_ID
ITEM
STATUS
HAVE
ORDER_ID
ORDER_DATE
ORDER
TOTAL_UNIT
TOTAL_PRICE
SUPPLIER
SUPPLIES
TAKEN BY
EMPLOYEE
S_NAME
S_ADDRESS
SUPPLIER_ID
E_NAME
EMPLOYEE_ID

Relational Model

- Customer (C_name, C_address, contact_no, customer_ID)
- Supplier (S_name, supplier_ID, S_address)
- Inventory(Drug_id, Drug_name, unit, price)
- Employee(Employee_name, Employee_ID)
- Orders (order_id, order_date, quantity, total_price, customer_id*, Employee_ID*)
- Item (item_id, expiry, status, supplier_ID*, Drug_id*, order_ID *)

IV. Normalization

Normalization is a systematic approach of decomposing tables to eliminate data redundancy(repetition) and undesirable characteristics like Insertion, Update and Deletion Anomalies. It is a multi-step process that puts data into tabular form, removing duplicated data from the relation tables.

Database normalization is the process of restructuring a relational database in accordance with a series of so-called normal forms in order to reduce data redundancy and improve data integrity. Normalization is a database design technique which organizes tables in a manner that reduces redundancy and dependency of data. It divides larger tables to smaller tables and links them using relationships

The objectives of normalization were stated as follows by Codd:

To free the collection of relations from undesirable insertion, update and deletion dependencies;
To reduce the need for restructuring the collection of relations, as new types of data are introduced, and thus increase the life span of application programs;
To make the relational model more informative to users;
To make the collection of relations neutral to the query statistics, where these statistics are liable to change as time goes by.

When an attempt is made to modify (update, insert into, or delete from) a relation, the following undesirable side-effects may arise in relations that have not been sufficiently normalized:

The most commonly used normal forms are:

First normal form(1NF) Second normal
form(2NF) Third normal form(3NF)
Boyce & Codd normal form (BCNF)

First normal form (1NF)

1NF (First Normal Form) Rules

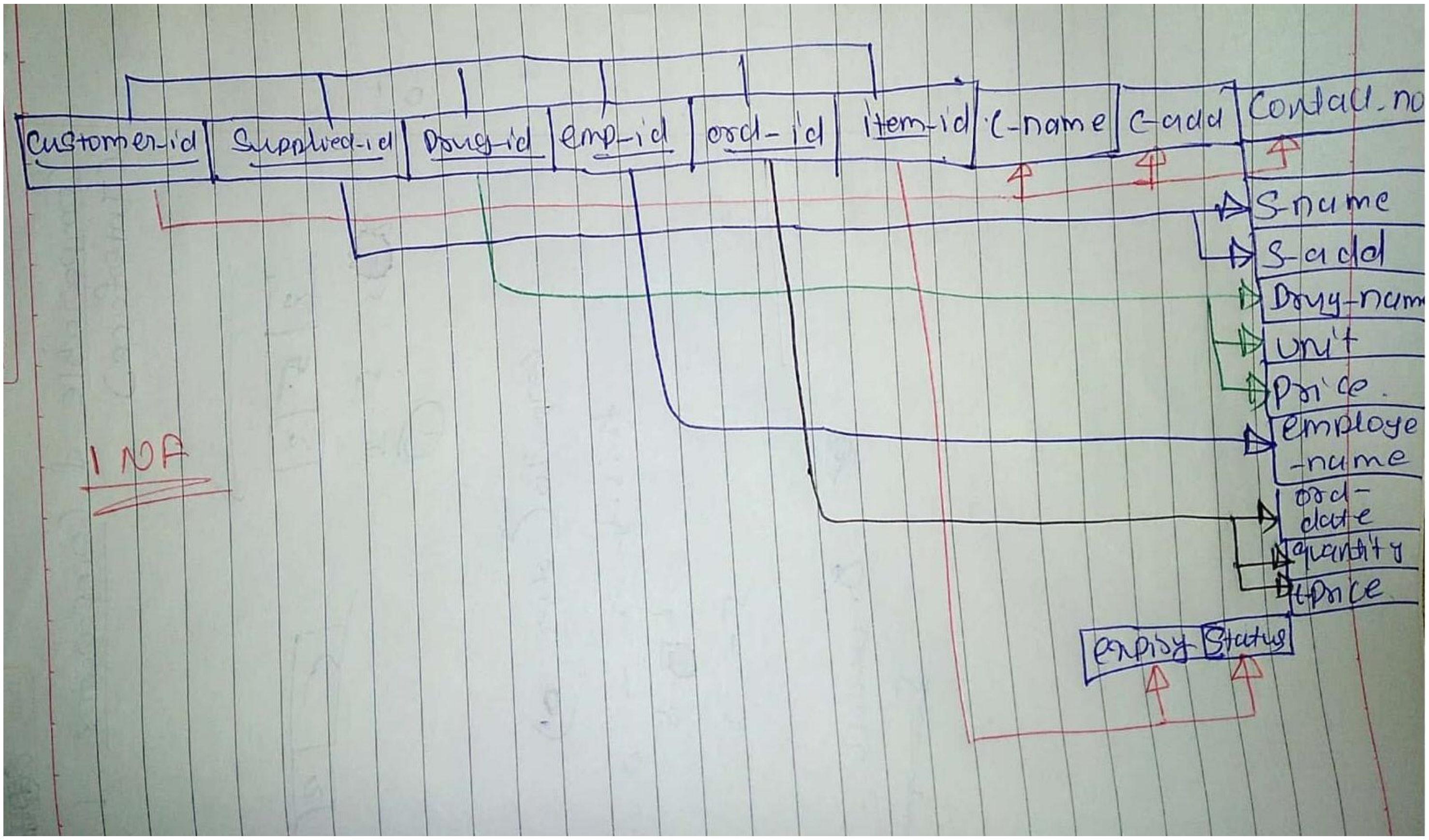
Each table cell should contain a single value. Each record needs to be unique.

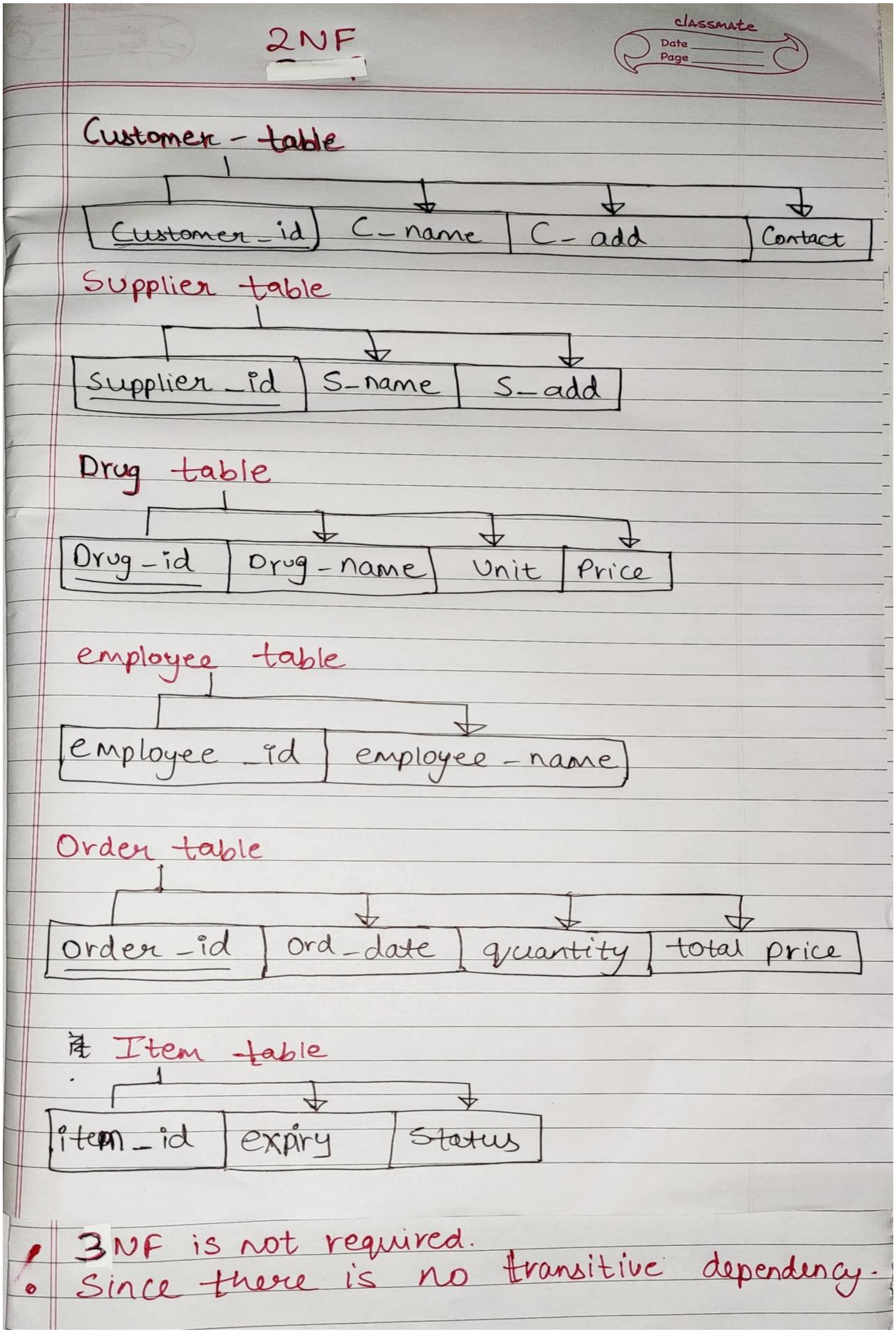
All the tables are in first normal form because no other entity has multivalued attributes

Second normal form (2NF)

2NF (Second Normal Form) Rules Rule 1- Be in

1NF
Rule 2- Single Column Primary Key. (No non-prime attribute is dependent on the proper subset of any candidate key of table.) An attribute that is not part of any candidate key is known as non-prime attribute.


Customer-id
Supplier-id
Drug-id
emp-id
ord-id
item-id
C-name
C-add
Contact-no
S-name
S-add
Drug-nam
unit
Price.
employe
-name
ord-
date
quantity
tprice
expiry
Status
1 NF



2NF

Customer - table

Customer_id	C_name	C_add	Contact

Supplier table

supplier_id	S_name	S_add

Drug table

Drug_id	Drug_name	Unit	Price

employee table

employee_id	employee_name

Order table

order_id	ord_date	quantity	total price

Item table

item_id	expiry	Status

3NF is not required.

Since there is no transitive dependency.

Third Normal Form (3NF)

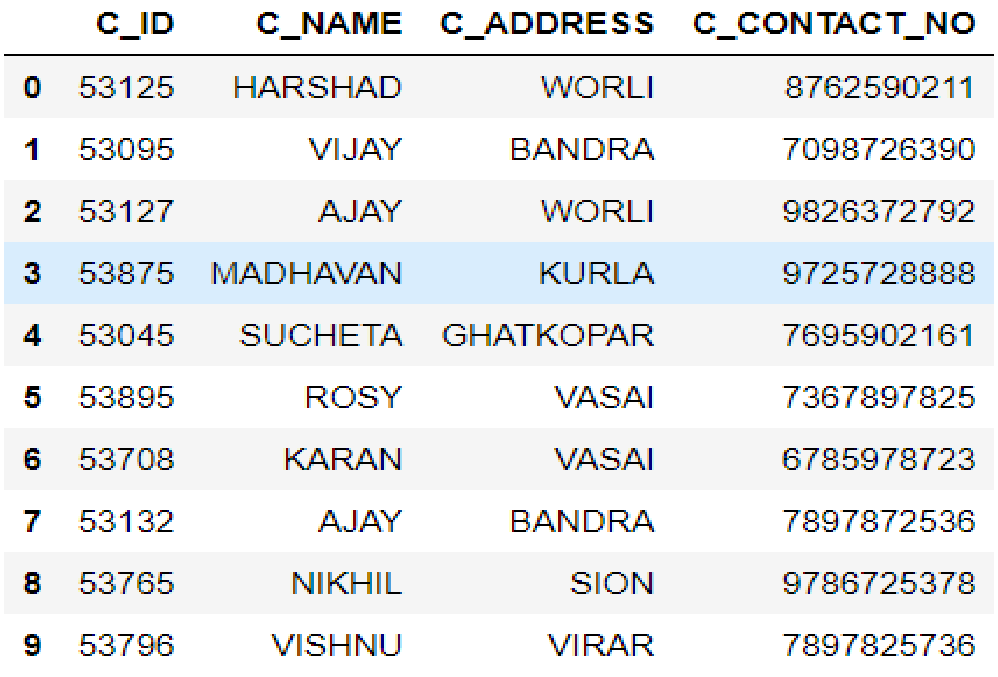
3NF (Third Normal Form) not required

Has no transitive functional dependencies.

TABLES

CUSTOMER

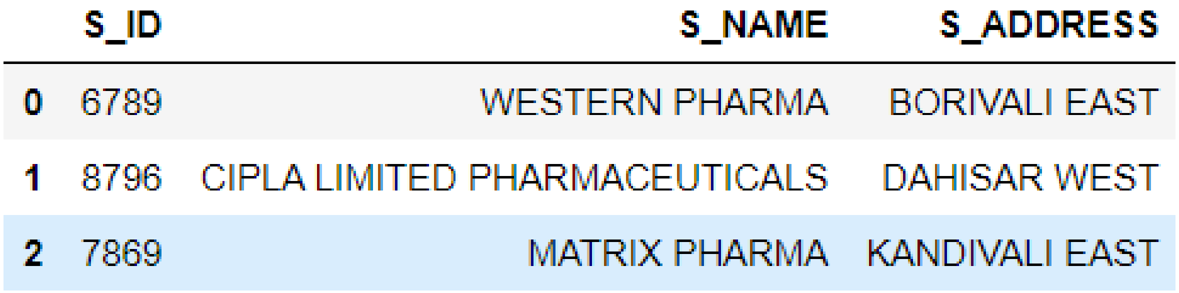
```
CREATE TABLE IF NOT EXISTS CUSTOMER
(C_ID       INT PRIMARY KEY      NOT NULL,
C_NAME      TEXT                 NOT NULL,
C_ADDRESS   TEXT                 NOT NULL,
C_CONTACT_NO INT                 NOT NULL)
```



	C_ID	C_NAME	C_ADDRESS	C_CONTACT_NO
0	53125	HARSHAD	WORLI	8762590211
1	53095	VIJAY	BANDRA	7098726390
2	53127	AJAY	WORLI	9826372792
3	53875	MADHAVAN	KURLA	9725728888
4	53045	SUCHETA	GHATKOPAR	7695902161
5	53895	ROSY	VASAI	7367897825
6	53708	KARAN	VASAI	6785978723
7	53132	AJAY	BANDRA	7897872536
8	53765	NIKHIL	SION	9786725378
9	53796	VISHNU	VIRAR	7897825736

SUPPLIER

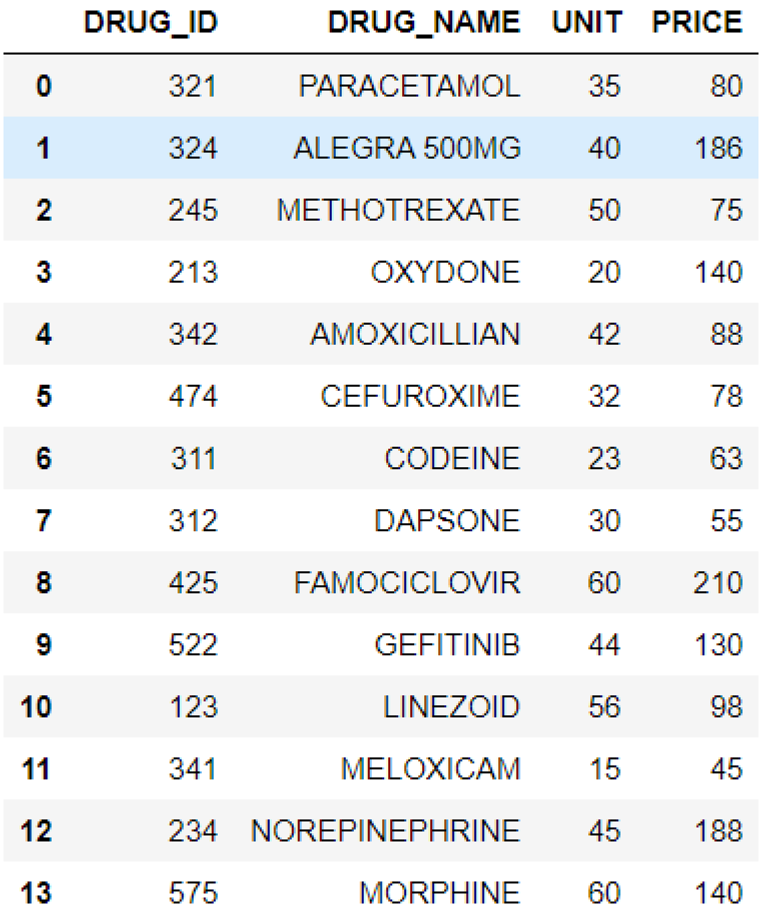
```
CREATE TABLE IF NOT EXISTS SUPPLIER
(S_ID       INT PRIMARY KEY       NOT NULL,
S_NAME      TEXT              NOT NULL,
S_ADDRESS   TEXT                  NOT NULL)
```



	S_ID	S_NAME	S_ADDRESS
0	6789	WESTERN PHARMA	BORIVALI EAST
1	8796	CIPLA LIMITED PHARMACEUTICALS	DAHISAR WEST
2	7869	MATRIX PHARMA	KANDIVALI EAST

INVENTORY

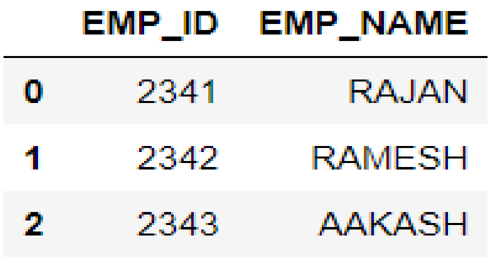
```
CREATE TABLE IF NOT EXISTS INVENTORY
(DRUG_ID       INT PRIMARY KEY     NOT NULL,
DRUG_NAME      TEXT                NOT NULL,
UNIT           INT                 NOT NULL,
PRICE          INT                 NOT NULL)
```



	DRUG_ID	DRUG_NAME	UNIT	PRICE
0	321	PARACETAMOL	35	80
1	324	ALEGRA 500MG	40	186
2	245	METHOTREXATE	50	75
3	213	OXYDONE	20	140
4	342	AMOXICILLIAN	42	88
5	474	CEFUROXIME	32	78
6	311	CODEINE	23	63
7	312	DAPSONE	30	55
8	425	FAMOCICLOVIR	60	210
9	522	GEFITINIB	44	130
10	123	LINEZOID	56	98
11	341	MELOXICAM	15	45
12	234	NOREPINEPHRINE	45	188
13	575	MORPHINE	60	140

EMPLOYEE

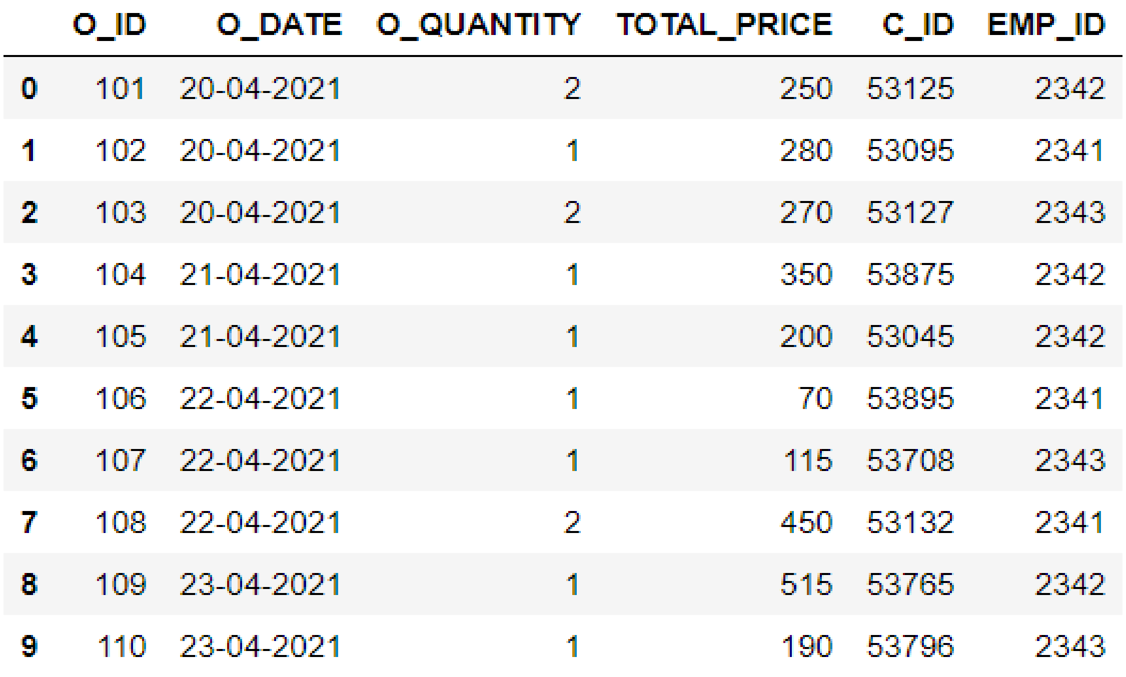
```
CREATE TABLE IF NOT EXISTS EMPLOYEE
(EMP_ID      INT PRIMARY KEY      NOT NULL,
EMP_NAME     TEXT                 NOT NULL)
```



	EMP_ID	EMP_NAME
0	2341	RAJAN
1	2342	RAMESH
2	2343	AAKASH

ORDERS

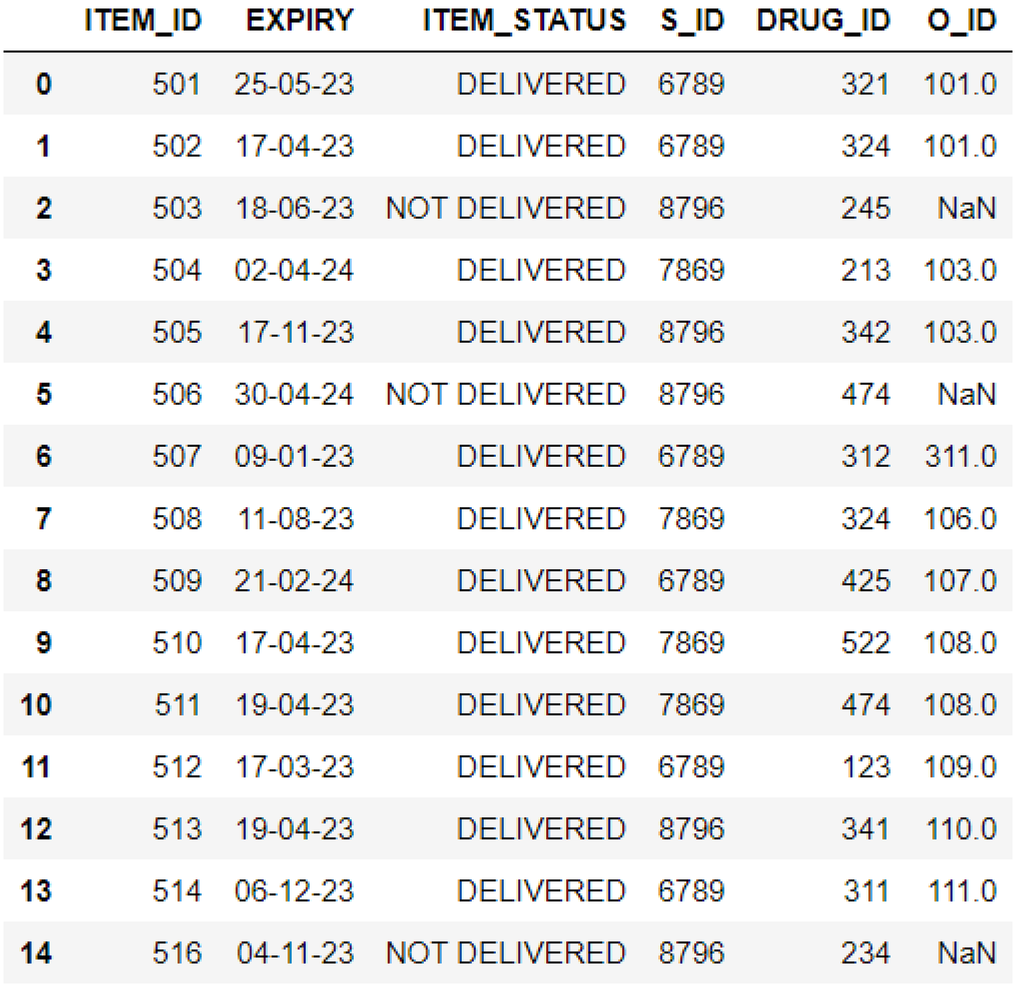
```
CREATE TABLE IF NOT EXISTS ORDERS
(O_ID              INT PRIMARY KEY       NOT NULL,
O_DATE            TEXT                   NOT NULL,
O_QUANTITY  INT                    NOT NULL,
TOTAL_PRICE   INT                    NOT NULL,
C_ID                INT                    NOT NULL,
EMP_ID            INT                    NOT NULL,
FOREIGN KEY (C_ID) REFERENCES CUSTOMER,
FOREIGN KEY (EMP_ID) REFERENCES EMPLOYEE)
```



	O_ID	O_DATE	O_QUANTITY	TOTAL_PRICE	C_ID	EMP_ID
0	101	20-04-2021	2	250	53125	2342
1	102	20-04-2021	1	280	53095	2341
2	103	20-04-2021	2	270	53127	2343
3	104	21-04-2021	1	350	53875	2342
4	105	21-04-2021	1	200	53045	2342
5	106	22-04-2021	1	70	53895	2341
6	107	22-04-2021	1	115	53708	2343
7	108	22-04-2021	2	450	53132	2341
8	109	23-04-2021	1	515	53765	2342
9	110	23-04-2021	1	190	53796	2343

ITEMS

```
CREATE TABLE IF NOT EXISTS ITEM
(ITEM_ID      INT PRIMARY KEY      NOT NULL,
EXPIRY           TEXT              NOT NULL,
ITEM_STATUS  TEXT           NOT NULL,
S_ID             INT               NOT NULL,
DRUG_ID       INT                NOT NULL,
O_ID             INT,
FOREIGN KEY (S_ID) REFERENCES SUPPLIER,
FOREIGN KEY (DRUG_ID) REFERENCES INVENTORY,
FOREIGN KEY (O_ID) REFERENCES ORDERS)
```



	ITEM_ID	EXPIRY	ITEM_STATUS	S_ID	DRUG_ID	O_ID
0	501	25-05-23	DELIVERED	6789	321	101.0
1	502	17-04-23	DELIVERED	6789	324	101.0
2	503	18-06-23	NOT DELIVERED	8796	245	NaN
3	504	02-04-24	DELIVERED	7869	213	103.0
4	505	17-11-23	DELIVERED	8796	342	103.0
5	506	30-04-24	NOT DELIVERED	8796	474	NaN
6	507	09-01-23	DELIVERED	6789	312	311.0
7	508	11-08-23	DELIVERED	7869	324	106.0
8	509	21-02-24	DELIVERED	6789	425	107.0
9	510	17-04-23	DELIVERED	7869	522	108.0
10	511	19-04-23	DELIVERED	7869	474	108.0
11	512	17-03-23	DELIVERED	6789	123	109.0
12	513	19-04-23	DELIVERED	8796	341	110.0
13	514	06-12-23	DELIVERED	6789	311	111.0
14	516	04-11-23	NOT DELIVERED	8796	234	NaN

Insert-

CUSTOMER-

```
INSERT INTO CUSTOMER (C_ID, C_NAME, C_ADDRESS, C_CONTACT_NO)
VALUES(53125,'HARSHAD ', 'WORLI', 8762590211)

INSERT INTO CUSTOMER (C_ID, C_NAME, C_ADDRESS, C_CONTACT_NO)
VALUES(53095, 'VIJAY ', 'BANDRA', 7098726390)

INSERT INTO CUSTOMER (C_ID, C_NAME, C_ADDRESS, C_CONTACT_NO)
VALUES(53127,'AJAY ', 'WORLI', 9826372792)

INSERT INTO CUSTOMER (C_ID, C_NAME, C_ADDRESS, C_CONTACT_NO)
VALUES(53875,'MADHAVAN',  'KURLA', 9725728888)

INSERT INTO CUSTOMER (C_ID, C_NAME, C_ADDRESS, C_CONTACT_NO)
VALUES(53045, 'SUCHETA ',  'GHATKOPAR', 7695902161)

INSERT INTO CUSTOMER (C_ID, C_NAME, C_ADDRESS, C_CONTACT_NO)
VALUES( 53895, 'ROSY',  'VASAI', 7367897825)

INSERT INTO CUSTOMER (C_ID, C_NAME, C_ADDRESS, C_CONTACT_NO)
VALUES(53708, 'KARAN', 'VASAI', 6785978723)

INSERT INTO CUSTOMER (C_ID, C_NAME, C_ADDRESS, C_CONTACT_NO)
VALUES(53132, 'AJAY ', 'BANDRA', 7897872536)

INSERT INTO CUSTOMER (C_ID, C_NAME, C_ADDRESS, C_CONTACT_NO)
VALUES(53765, 'NIKHIL', 'SION', 9786725378)

INSERT INTO CUSTOMER (C_ID, C_NAME, C_ADDRESS, C_CONTACT_NO)
 VALUES(53796, 'VISHNU', 'VIRAR', 7897825736)
```

SUPPLIER:

```
INSERT INTO SUPPLIER(S_ID, S_NAME, S_ADDRESS)
VALUES( 6789 ,'WESTERN PHARMA','BORIVALI EAST')

INSERT INTO SUPPLIER(S_ID, S_NAME, S_ADDRESS)
VALUES( 8796 ,'CIPLA LIMITED PHARMACEUTICALS','DAHISAR WEST')

INSERT INTO SUPPLIER(S_ID, S_NAME, S_ADDRESS)
VALUES( 7869 ,'MATRIX PHARMA','KANDIVALI EAST' )
```

INVENTORY:

```
INSERT INTO INVENTORY(DRUG_ID, DRUG_NAME, UNIT, PRICE  )
VALUES(321, 'PARACETAMOL', 35 , 80 )

INSERT INTO INVENTORY(DRUG_ID, DRUG_NAME, UNIT, PRICE  )
VALUES(324, 'ALEGRA 500MG', 40, 186 )

INSERT INTO INVENTORY(DRUG_ID, DRUG_NAME, UNIT, PRICE  )
VALUES(245, 'METHOTREXATE', 50, 75)

INSERT INTO INVENTORY(DRUG_ID, DRUG_NAME, UNIT, PRICE  )
VALUES(213, 'OXYDONE', 20, 140)

INSERT INTO INVENTORY(DRUG_ID, DRUG_NAME, UNIT, PRICE  )
VALUES(342, 'AMOXICILLIAN', 42, 88)

INSERT INTO INVENTORY(DRUG_ID, DRUG_NAME, UNIT, PRICE  )
VALUES(474, 'CEFUROXIME', 32, 78)

INSERT INTO INVENTORY(DRUG_ID, DRUG_NAME, UNIT, PRICE  )
VALUES(311, 'CODEINE', 23, 63 )

INSERT INTO INVENTORY(DRUG_ID, DRUG_NAME, UNIT, PRICE  )
VALUES(312, 'DAPSONE', 30, 55)

INSERT INTO INVENTORY(DRUG_ID, DRUG_NAME, UNIT, PRICE  )
VALUES(425, 'FAMOCICLOVIR', 60, 210)

INSERT INTO INVENTORY(DRUG_ID, DRUG_NAME, UNIT, PRICE  )
VALUES(522, 'GEFITINIB', 44, 130 )

INSERT INTO INVENTORY(DRUG_ID, DRUG_NAME, UNIT, PRICE  )
VALUES(123, 'LINEZOID', 56, 98)

INSERT INTO INVENTORY(DRUG_ID, DRUG_NAME, UNIT, PRICE  )
VALUES(341, 'MELOXICAM', 15, 45)

INSERT INTO INVENTORY(DRUG_ID, DRUG_NAME, UNIT, PRICE  )
VALUES(234, 'NOREPINEPHRINE', 45, 188 )

INSERT INTO INVENTORY(DRUG_ID, DRUG_NAME, UNIT, PRICE  )
VALUES(575, 'MORPHINE', 60, 140 )
```

EMPLOYEE-

```
INSERT INTO EMPLOYEE( EMP_ID, EMP_NAME)
VALUES( 2341, 'RAJAN')

INSERT INTO EMPLOYEE( EMP_ID, EMP_NAME)
VALUES(2342, 'RAMESH' )

INSERT INTO EMPLOYEE( EMP_ID, EMP_NAME)
VALUES(2343, 'AAKASH' )
```

ORDERS-

```
INSERT INTO ORDERS( O_ID, O_DATE, O_QUANTITY, TOTAL_PRICE, C_ID, EMP_ID)
VALUES(101, '20-04-2021', 2, 250, 53125, 2342 )

INSERT INTO ORDERS( O_ID, O_DATE, O_QUANTITY, TOTAL_PRICE, C_ID, EMP_ID)
VALUES(102, '20-04-2021', 1, 280, 53095, 2341 )

INSERT INTO ORDERS( O_ID, O_DATE, O_QUANTITY, TOTAL_PRICE, C_ID, EMP_ID)
VALUES(103, '20-04-2021', 2, 270, 53127, 2343 )

INSERT INTO ORDERS( O_ID, O_DATE, O_QUANTITY, TOTAL_PRICE, C_ID, EMP_ID)
VALUES(104, '21-04-2021', 1, 350, 53875, 2342 )

INSERT INTO ORDERS( O_ID, O_DATE, O_QUANTITY, TOTAL_PRICE, C_ID, EMP_ID)
VALUES(105, '21-04-2021', 1, 200, 53045, 2342 )

INSERT INTO ORDERS( O_ID, O_DATE, O_QUANTITY, TOTAL_PRICE, C_ID, EMP_ID)
VALUES(106, '22-04-2021', 1, 70, 53895, 2341)

INSERT INTO ORDERS( O_ID, O_DATE, O_QUANTITY, TOTAL_PRICE, C_ID, EMP_ID)
VALUES(107, '22-04-2021', 1, 115, 53708, 2343 )

INSERT INTO ORDERS( O_ID, O_DATE, O_QUANTITY, TOTAL_PRICE, C_ID, EMP_ID)
VALUES(108, '22-04-2021', 2, 450, 53132, 2341 )

INSERT INTO ORDERS( O_ID, O_DATE, O_QUANTITY, TOTAL_PRICE, C_ID, EMP_ID)
VALUES(109, '23-04-2021', 1, 515, 53765, 2342 )

INSERT INTO ORDERS( O_ID, O_DATE, O_QUANTITY, TOTAL_PRICE, C_ID, EMP_ID)
VALUES(110, '23-04-2021', 1, 190, 53796, 2343 )
```

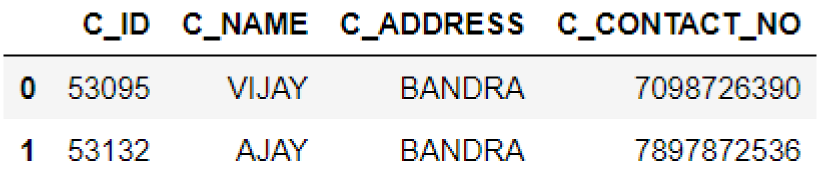
ITEM:

```
INSERT INTO ITEM( ITEM_ID, EXPIRY, ITEM_STATUS, S_ID, DRUG_ID, O_ID)
VALUES(501, '25-05-23', 'DELIVERED', 6789, 321, 101 )

INSERT INTO ITEM( ITEM_ID, EXPIRY, ITEM_STATUS, S_ID, DRUG_ID, O_ID)
VALUES(502, '17-04-23', 'DELIVERED', 6789, 324, 101 )
INSERT INTO ITEM( ITEM_ID, EXPIRY, ITEM_STATUS, S_ID, DRUG_ID)
VALUES(503, '18-06-23', 'NOT DELIVERED', 8796, 245 )

INSERT INTO ITEM( ITEM_ID, EXPIRY, ITEM_STATUS, S_ID, DRUG_ID, O_ID)
VALUES(504, '02-04-24', 'DELIVERED', 7869, 213, 103 )

INSERT INTO ITEM( ITEM_ID, EXPIRY, ITEM_STATUS, S_ID, DRUG_ID, O_ID)
VALUES(505, '17-11-23', 'DELIVERED', 8796, 342, 103 )

INSERT INTO ITEM( ITEM_ID, EXPIRY, ITEM_STATUS, S_ID, DRUG_ID)
VALUES(506, '30-04-24', 'NOT DELIVERED', 8796, 474)

INSERT INTO ITEM( ITEM_ID, EXPIRY, ITEM_STATUS, S_ID, DRUG_ID, O_ID)
VALUES(507, '09-01-23', 'DELIVERED', 6789, 312, 311 )

INSERT INTO ITEM( ITEM_ID, EXPIRY, ITEM_STATUS, S_ID, DRUG_ID, O_ID)
VALUES(508, '11-08-23', 'DELIVERED', 7869, 324, 106 )

INSERT INTO ITEM( ITEM_ID, EXPIRY, ITEM_STATUS, S_ID, DRUG_ID, O_ID)
VALUES(509, '21-02-24', 'DELIVERED', 6789, 425, 107 )

INSERT INTO ITEM( ITEM_ID, EXPIRY, ITEM_STATUS, S_ID, DRUG_ID, O_ID)
VALUES(510, '17-04-23', 'DELIVERED', 7869, 522, 108 )

INSERT INTO ITEM( ITEM_ID, EXPIRY, ITEM_STATUS, S_ID, DRUG_ID, O_ID)
VALUES(511, '19-04-23', 'DELIVERED', 7869, 474, 108)

INSERT INTO ITEM( ITEM_ID, EXPIRY, ITEM_STATUS, S_ID, DRUG_ID, O_ID)
VALUES(512, '17-03-23', 'DELIVERED', 6789, 123, 109 )

INSERT INTO ITEM( ITEM_ID, EXPIRY, ITEM_STATUS, S_ID, DRUG_ID, O_ID)
VALUES(513, '19-04-23', 'DELIVERED', 8796, 341, 110 )

INSERT INTO ITEM( ITEM_ID, EXPIRY, ITEM_STATUS, S_ID, DRUG_ID, O_ID)
VALUES(514, '06-12-23', 'DELIVERED', 6789, 311, 111 )

INSERT INTO ITEM( ITEM_ID, EXPIRY, ITEM_STATUS, S_ID, DRUG_ID, O_ID)
VALUES(515, '28-10-24', 'DELIVERED', 7869, 575, 112 )

INSERT INTO ITEM( ITEM_ID, EXPIRY, ITEM_STATUS, S_ID, DRUG_ID)
VALUES(516, '04-11-23', 'NOT DELIVERED', 8796, 234)
```

Queries-

1) WAQ TO DISPLAY CUSTOMER DETAILS WHO ARE FROM BANDRA.

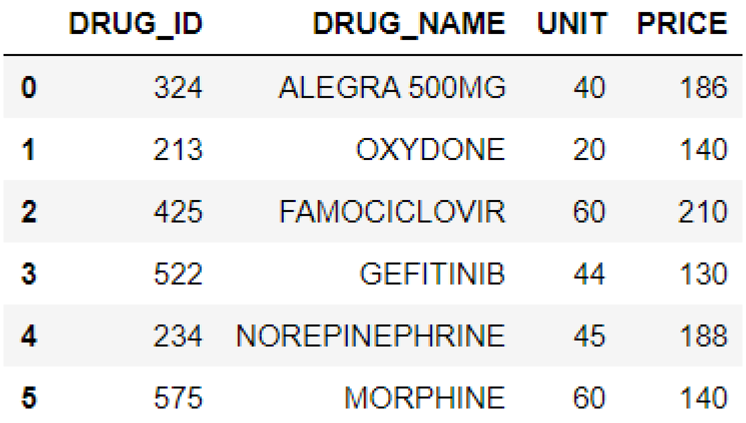
SELECT *
FROM CUSTOMER
WHERE C_ADDRESS = 'BANDRA'



	C_ID	C_NAME	C_ADDRESS	C_CONTACT_NO
0	53095	VIJAY	BANDRA	7098726390
1	53132	AJAY	BANDRA	7897872536

2) WAQ TO DISPLAY THE DETAILS OF DRUGS WHOSE PRICE IS GREATER THAN 100.

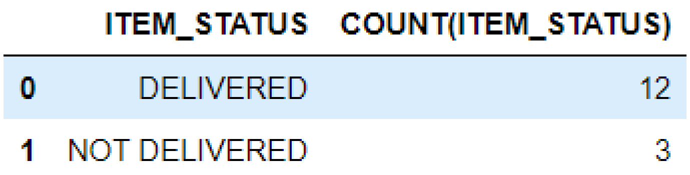
SELECT *
FROM INVENTORY
WHERE PRICE > 100



	DRUG_ID	DRUG_NAME	UNIT	PRICE
0	324	ALEGRA 500MG	40	186
1	213	OXYDONE	20	140
2	425	FAMOCICLOVIR	60	210
3	522	GEFITINIB	44	130
4	234	NOREPINEPHRINE	45	188
5	575	MORPHINE	60	140

3) WAQ TO DISPLAY THE NUMBER OF ITEMS WHICH HAS BEEN DELIVERED.

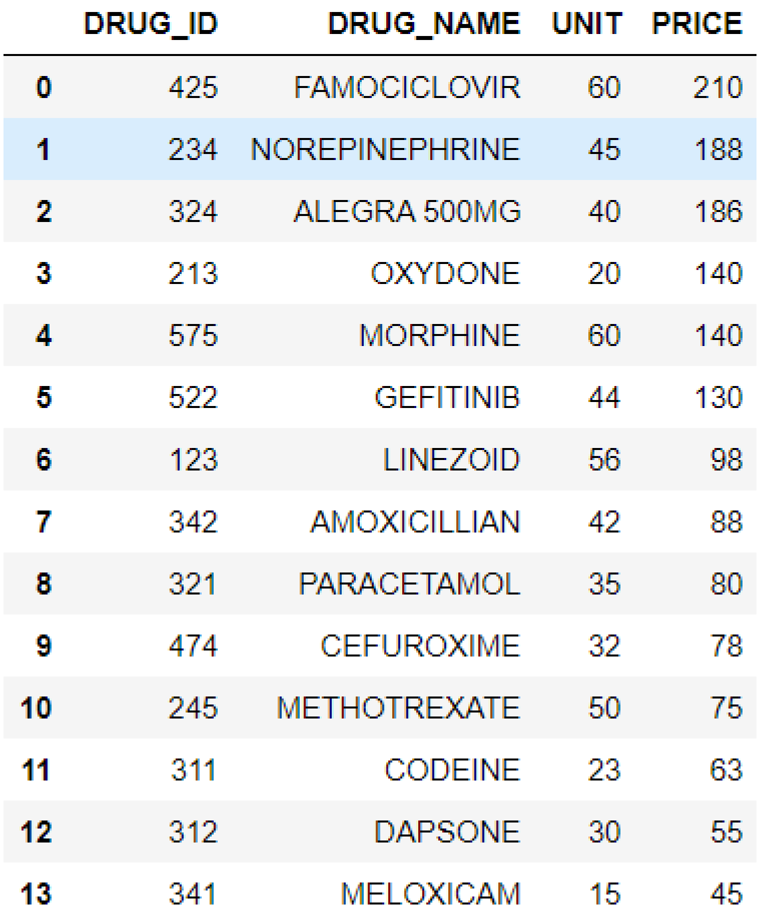
SELECT ITEM_STATUS, COUNT(ITEM_STATUS)
FROM ITEM
GROUP BY ITEM_STATUS



	ITEM_STATUS	COUNT(ITEM_STATUS)
0	DELIVERED	12
1	NOT DELIVERED	3

4) WAQ TO DISPLAY DETAILS WITH UNIT PRICE IN DESCENDING ORDER.

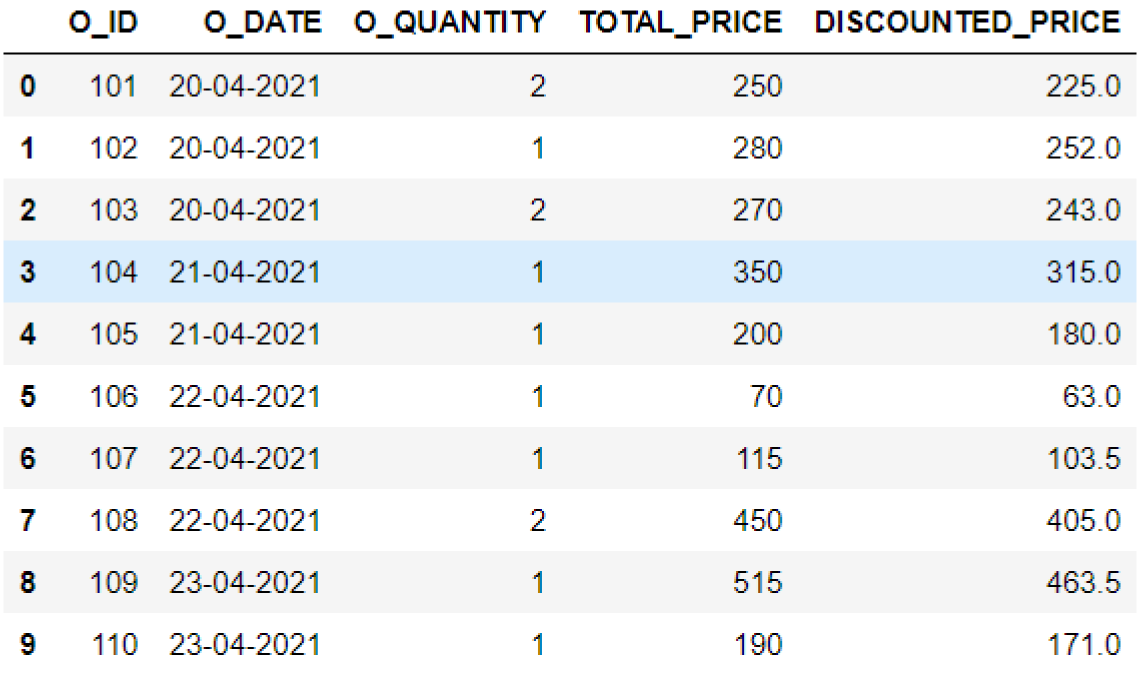
```
SELECT *
FROM INVENTORY
ORDER BY PRICE DESC
```



	DRUG_ID	DRUG_NAME	UNIT	PRICE
0	425	FAMOCICLOVIR	60	210
1	234	NOREPINEPHRINE	45	188
2	324	ALEGRA 500MG	40	186
3	213	OXYDONE	20	140
4	575	MORPHINE	60	140
5	522	GEFITINIB	44	130
6	123	LINEZOID	56	98
7	342	AMOXICILLIAN	42	88
8	321	PARACETAMOL	35	80
9	474	CEFUROXIME	32	78
10	245	METHOTREXATE	50	75
11	311	CODEINE	23	63
12	312	DAPSONE	30	55
13	341	MELOXICAM	15	45

5) WAQ TO DISPLAY DISCOUNTED PRICE ALONG WITH ALL ATTRIBUTES OF ORDERS.

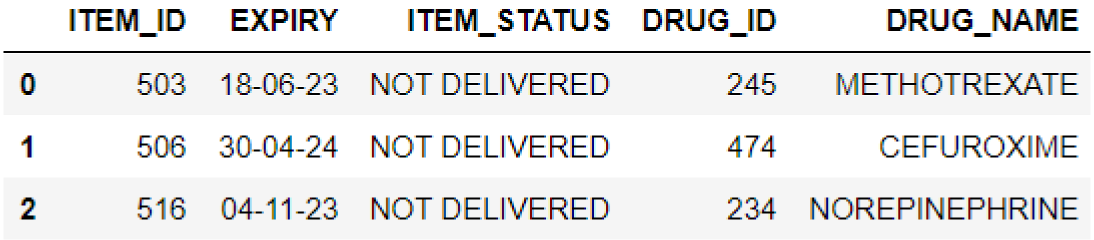
```
SELECT O_ID, O_DATE, O_QUANTITY, TOTAL_PRICE, TOTAL_PRICE - (TOTAL_PRICE * 0.1)
AS DISCOUNTED_PRICE
FROM ORDERS
```



	O_ID	O_DATE	O_QUANTITY	TOTAL_PRICE	DISCOUNTED_PRICE
0	101	20-04-2021	2	250	225.0
1	102	20-04-2021	1	280	252.0
2	103	20-04-2021	2	270	243.0
3	104	21-04-2021	1	350	315.0
4	105	21-04-2021	1	200	180.0
5	106	22-04-2021	1	70	63.0
6	107	22-04-2021	1	115	103.5
7	108	22-04-2021	2	450	405.0
8	109	23-04-2021	1	515	463.5
9	110	23-04-2021	1	190	171.0

6) WAQ TO FIND THE DETAILS DRUG WHICH HAS NOT BEEN DELIVERED.

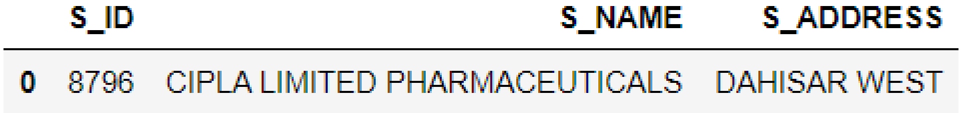
```
SELECT  U.ITEM_ID, U.EXPIRY, U.ITEM_STATUS, I.DRUG_ID, I.DRUG_NAME
FROM INVENTORY AS I NATURAL JOIN ITEM AS U
WHERE U.ITEM_STATUS = 'NOT DELIVERED'
```



	ITEM_ID	EXPIRY	ITEM_STATUS	DRUG_ID	DRUG_NAME
0	503	18-06-23	NOT DELIVERED	245	METHOTREXATE
1	506	30-04-24	NOT DELIVERED	474	CEFUROXIME
2	516	04-11-23	NOT DELIVERED	234	NOREPINEPHRINE

7) WAQ TO DISPLAY THE DETAILS OF THE SUPPLIER WHO SUPPLIES THE ITEMS WHICH HAVE LESS THAN 20 UNITS IN THE INVENTORY.

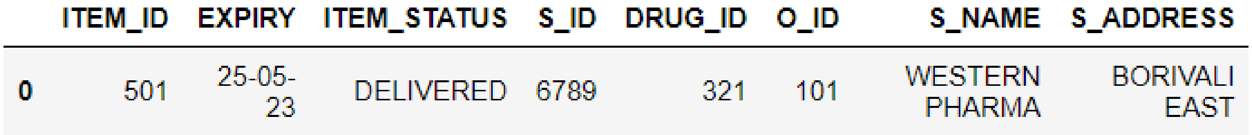
```
SELECT *
FROM SUPPLIER
WHERE S_ID IN
        (SELECT  S_ID
         FROM  ITEM
         WHERE DRUG_ID IN
                    (SELECT  DRUG_ID
                     FROM  INVENTORY
                     WHERE  UNIT  <  20))
```



	S_ID	S_NAME	S_ADDRESS
0	8796	CIPLA LIMITED PHARMACEUTICALS	DAHISAR WEST

8) WAQ TO DISPLAY ITEM DETAILS SUPPLIED BY THE SUPPLIER HAVING SUPPLIER ID 8796.

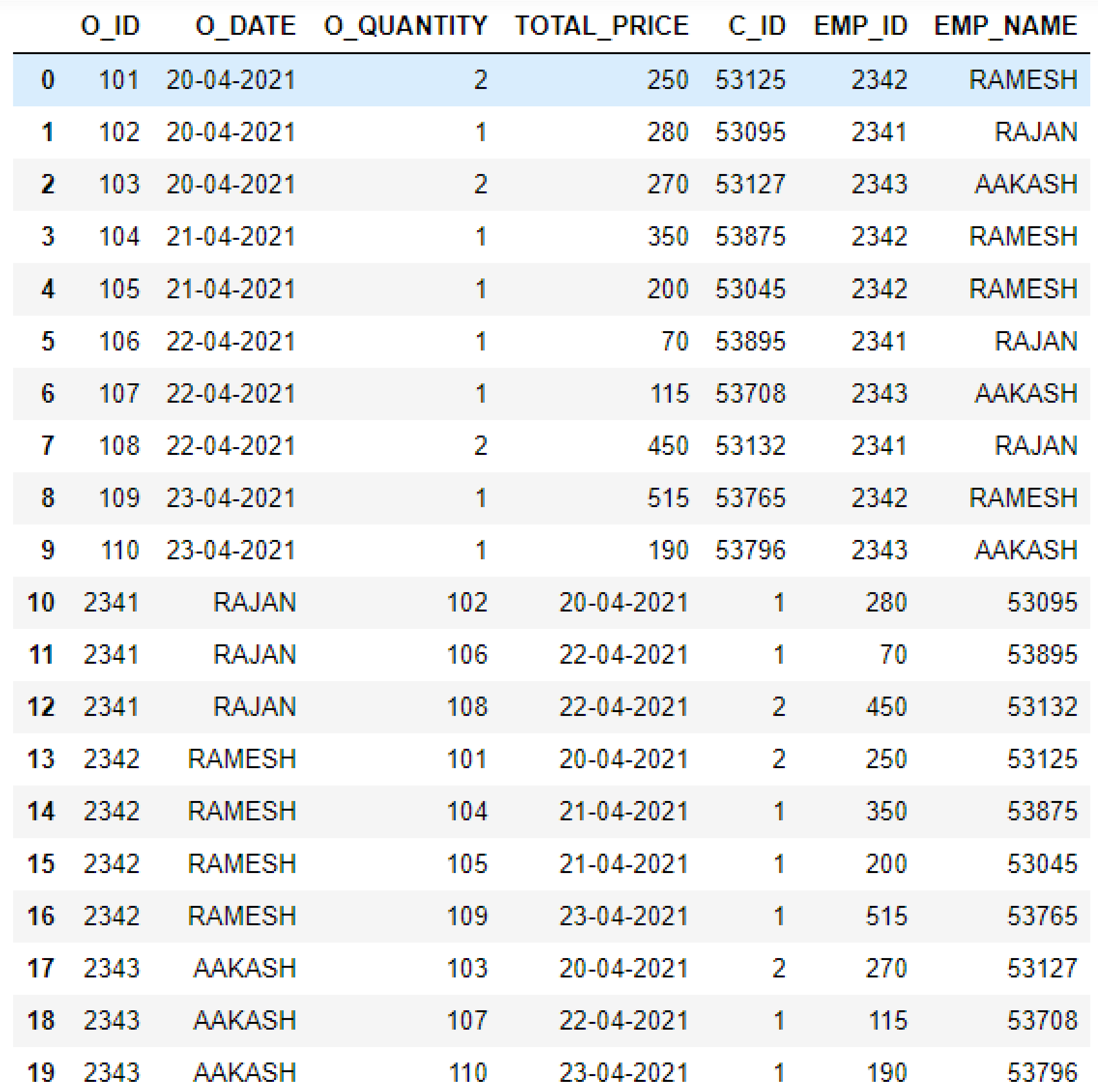
```
SELECT  *
FROM ITEM AS I NATURAL INNER JOIN SUPPLIER AS S
GROUP BY S_NAME
HAVING  S_ID = 6789
```



	ITEM_ID	EXPIRY	ITEM_STATUS	S_ID	DRUG_ID	O_ID	S_NAME	S_ADDRESS
0	501	25-05-23	DELIVERED	6789	321	101	WESTERN PHARMA	BORIVALI EAST

9) WAQ to Simulate Full Outer Join between ORDERS and EMPLOYEE.

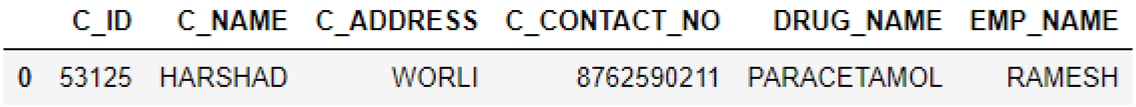
```
SELECT  *
FROM ORDERS AS O NATURAL LEFT OUTER JOIN EMPLOYEE AS E

UNION ALL

SELECT  *
FROM   EMPLOYEE AS E NATURAL LEFT OUTER JOIN ORDERS AS O
```



	O_ID	O_DATE	O_QUANTITY	TOTAL_PRICE	C_ID	EMP_ID	EMP_NAME
0	101	20-04-2021	2	250	53125	2342	RAMESH
1	102	20-04-2021	1	280	53095	2341	RAJAN
2	103	20-04-2021	2	270	53127	2343	AAKASH
3	104	21-04-2021	1	350	53875	2342	RAMESH
4	105	21-04-2021	1	200	53045	2342	RAMESH
5	106	22-04-2021	1	70	53895	2341	RAJAN
6	107	22-04-2021	1	115	53708	2343	AAKASH
7	108	22-04-2021	2	450	53132	2341	RAJAN
8	109	23-04-2021	1	515	53765	2342	RAMESH
9	110	23-04-2021	1	190	53796	2343	AAKASH
10	2341	RAJAN	102	20-04-2021	1	280	53095
11	2341	RAJAN	106	22-04-2021	1	70	53895
12	2341	RAJAN	108	22-04-2021	2	450	53132
13	2342	RAMESH	101	20-04-2021	2	250	53125
14	2342	RAMESH	104	21-04-2021	1	350	53875
15	2342	RAMESH	105	21-04-2021	1	200	53045
16	2342	RAMESH	109	23-04-2021	1	515	53765
17	2343	AAKASH	103	20-04-2021	2	270	53127
18	2343	AAKASH	107	22-04-2021	1	115	53708
19	2343	AAKASH	110	23-04-2021	1	190	53796

10) WAQ TO DISPLAY THE CUSTOMER WHO HAS ORDERED PARACETAMOL FROM THE EMPLOYEE.

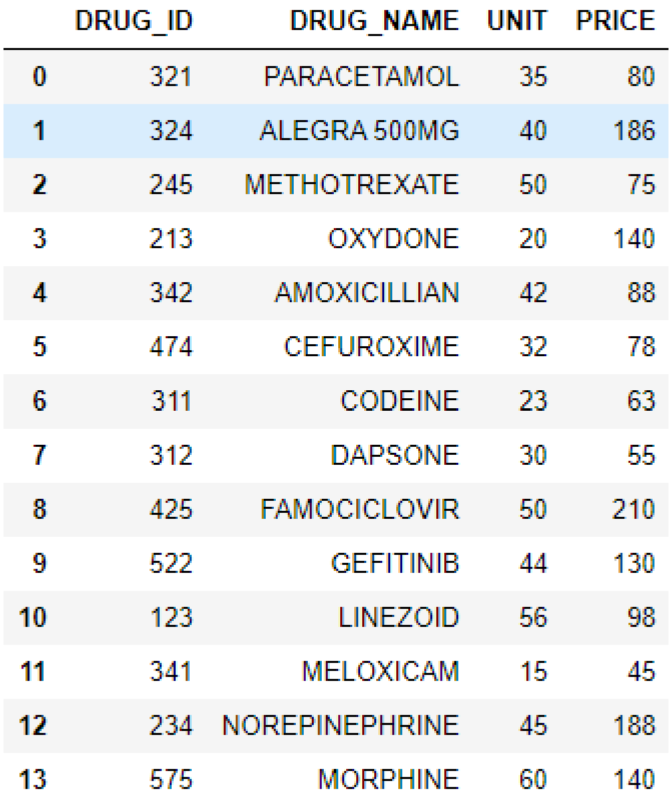
```
SELECT  C.C_ID, C.C_NAME, C.C_ADDRESS, C.C_CONTACT_NO, V.DRUG_NAME,
E.EMP_NAME
FROM  CUSTOMER AS C ,  ORDERS AS O,  ITEM AS I, INVENTORY AS V, EMPLOYEE AS E
WHERE I.O_ID = O.O_ID
AND      O.C_ID = C.C_ID
AND      I.DRUG_ID = V.DRUG_ID
AND      E.EMP_ID = O.EMP_ID
AND      V.DRUG_NAME = 'PARACETAMOL'
```



	C_ID	C_NAME	C_ADDRESS	C_CONTACT_NO	DRUG_NAME	EMP_NAME
0	53125	HARSHAD	WORLI	8762590211	PARACETAMOL	RAMESH

11) WAQ TO UPDATE THE UNIT OF FAMOCICLOVIR FROM INVENTORY.

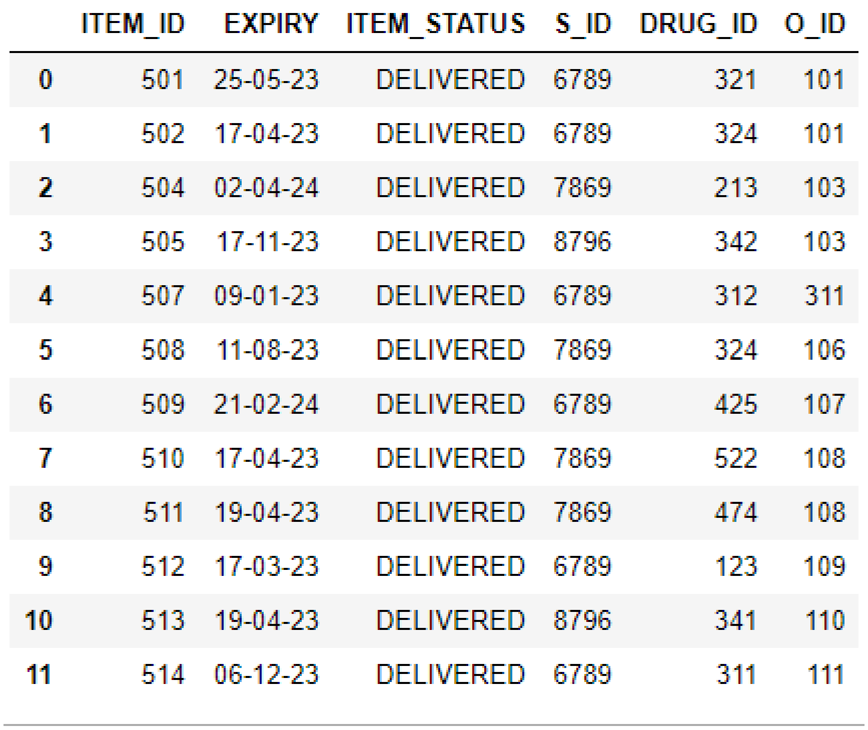
```
UPDATE INVENTORY
SET UNIT = 50
WHERE DRUG_ID = 425

SELECT *
FROM INVENTORY
```



	DRUG_ID	DRUG_NAME	UNIT	PRICE
0	321	PARACETAMOL	35	80
1	324	ALEGRA 500MG	40	186
2	245	METHOTREXATE	50	75
3	213	OXYDONE	20	140
4	342	AMOXICILLIAN	42	88
5	474	CEFUROXIME	32	78
6	311	CODEINE	23	63
7	312	DAPSONE	30	55
8	425	FAMOCICLOVIR	50	210
9	522	GEFITINIB	44	130
10	123	LINEZOID	56	98
11	341	MELOXICAM	15	45
12	234	NOREPINEPHRINE	45	188
13	575	MORPHINE	60	140

12) WAQ TO DELETE ITEM DETAILS WHOSE ITEM STATUS IS 'NOT DELIVERED'.

```
DELETE FROM ITEM
WHERE ITEM_STATUS = 'NOT DELIVERED'

SELECT *
FROM ITEM
```



	ITEM_ID	EXPIRY	ITEM_STATUS	S_ID	DRUG_ID	O_ID
0	501	25-05-23	DELIVERED	6789	321	101
1	502	17-04-23	DELIVERED	6789	324	101
2	504	02-04-24	DELIVERED	7869	213	103
3	505	17-11-23	DELIVERED	8796	342	103
4	507	09-01-23	DELIVERED	6789	312	311
5	508	11-08-23	DELIVERED	7869	324	106
6	509	21-02-24	DELIVERED	6789	425	107
7	510	17-04-23	DELIVERED	7869	522	108
8	511	19-04-23	DELIVERED	7869	474	108
9	512	17-03-23	DELIVERED	6789	123	109
10	513	19-04-23	DELIVERED	8796	341	110
11	514	06-12-23	DELIVERED	6789	311	111