**MYSQL Practical**

Create below tables and also add required primaary key, foreign key and constraints in it.

- User,

- Product,

- Order (with User ID, Order status and expected delivery date)

- Order detail (order id, product Ids)

1. Fetch all the User order list and include atleast following details in that.

- Customer name

- Product names

- Order Date

- Expected delivery date (in days, i.e. within X days)

2. Create summary report which provide information about

- All undelivered Orders

- 5 Most recent orders

- Top 5 active users (Users having most number of orders)

- Inactive users (Users who hasn’t done any order)

- Top 5 Most purchased products

- Most expensive and most chepest orders.

* **Creating Database:**

CREATE DATABASE db\_practical;

* **Using Database:**

USE db\_practical;

* **Creating USER Table:**

CREATE TABLE user (

`userid` INT(11) PRIMARY KEY NOT NULL,

`name` VARCHAR(20) NOT NULL,

`contact` INT(10) NOT NULL,

`address` VARCHAR(50),

`gender` CHAR(2) NOT NULL,

`nationality` VARCHAR(50)

);

* **Adding Auto-increment to primary Key**

ALTER TABLE `db\_practical`.`user`

CHANGE COLUMN `userid` `userid` INT NOT NULL AUTO\_INCREMENT ,

ADD UNIQUE INDEX `userid\_UNIQUE` (`userid` ASC) VISIBLE;

* **Inserting Data:**

INSERT INTO `user` (`name`,`contact`,`address`,`gender`,`nationality`) VALUES('Abhishek',1234567,'Mahidharpura Surat 395003','M','Indian') ;

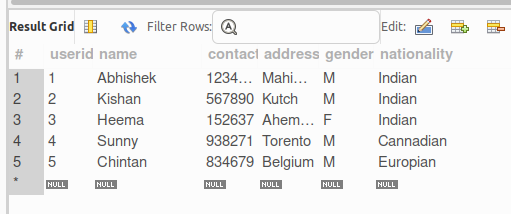
INSERT INTO `user`(`name`,`contact`,`address`,

`gender`,`nationality`) VALUES('Kishan',567890,'Kutch','M','Indian'),

('Heema',152637,'Ahemadabad','F','Indian'),

('Sunny',938271,'Torento','M','Cannadian'),

('Chintan',834679,'Belgium','M','Europian');



* **Creating PRODUCT Table:**

CREATE TABLE product (

`productid` INT PRIMARY KEY NOT NULL

AUTO\_INCREMENT UNIQUE,

`product\_name` VARCHAR(20) NOT NULL,

`manufacturerid` INT NOT NULL,

`price` FLOAT DEFAULT 0.0,

`quantity` INT DEFAULT 0,

`weight` FLOAT DEFAULT 0,

FOREIGN KEY (manufacturerid) REFERENCES user(userid)

ON UPDATE CASCADE

ON DELETE CASCADE

);

* **Inserting Data:**

INSERT INTO `product` (`product\_name`,`manufacturerid`,`price`,`quantity`,

`weight`) VALUES('ONE PLUS Nord CE 2',4,26500.00,10,3.75);

INSERT INTO `product` (`product\_name`,`manufacturerid`,`price`,`quantity`,

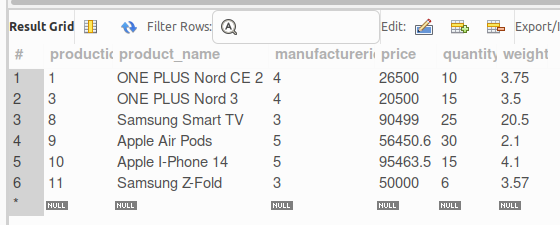
`weight`) VALUES('ONE PLUS Nord 3',4,20500.00,15,3.50),

('Samsung Smart TV',3,90499.00,25,20.50),

('Apple Air Pods',5,56450.55,30,2.10),

('Apple I-Phone 14',5,95463.55,15,4.10),

('Samsung Z-Fold',3,50000.00,6,3.57);



* **Creating ORDER Table:**

CREATE TABLE `order` (

`orderid` INT PRIMARY KEY NOT NULL

AUTO\_INCREMENT UNIQUE,

`userid` INT NOT NULL,

`order\_status` VARCHAR(25) DEFAULT "Pending",

`order\_date` DATE NOT NULL,

`expected\_delivery` DATE NOT NULL,

`quantity` INT DEFAULT 0,

FOREIGN KEY (userid) REFERENCES user(userid)

ON UPDATE CASCADE

ON DELETE CASCADE

);

* **Inserting Data:**

INSERT INTO `order`(`userid`,`order\_status`,`order\_date`,

`expected\_delivery`,`quantity`) VALUES (1,'Accepted',

'2022-04-16','2022-04-17',1);

INSERT INTO `order`(`userid`,`order\_status`,`order\_date`,

`expected\_delivery`,`quantity`)

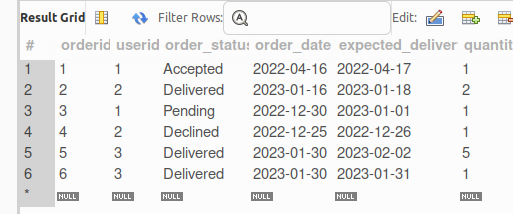
VALUES (2,'Delivered','2023-01-16','2023-01-16',2),

(1,'Pending','2022-12-30','2023-01-01',1),

(2,'Declined','2022-12-25','2022-12-26',1),

(3,'Delivered','2023-01-30','2023-01-30',5),

(3,'Delivered','2023-01-30','2023-01-30',1);



* **Creating ORDER\_DETAILS Table:**

CREATE TABLE `order\_details` (

`orderid` INT NOT NULL REFERENCES `order`(orderid)

ON UPDATE CASCADE

ON DELETE CASCADE,

`productid` INT NOT NULL REFERENCES `product`(productid)

ON UPDATE CASCADE

ON DELETE CASCADE,

`trans\_status` VARCHAR(30) DEFAULT 'Pending',

`delivery\_date` DATE);

* **Inserting Data:**

INSERT INTO `order\_details` (`orderid`,`productid`,`trans\_status`,`delivery\_date`)

VALUES (1,1,"Done",NULL),

(2,1,"Done","2023-01-18"),

(5,3,"Done","2023-01-30"),

(6,4,"Done","2023-02-01"),

(3,3,"Pending",NULL);

INSERT INTO order\_details(orderid,productid,trans\_status) VALUES(4,9,"Done");

* **Adding Amount Column**

ALTER TABLE `db\_practical`.`order\_details`

ADD COLUMN `amount` BIGINT NULL DEFAULT 0 AFTER `delivery\_date`;

* **Update amount column**

UPDATE order\_details SET amount = (SELECT p.price\*o.quantity FROM `order` o , `product` p WHERE o.orderid=1 AND p.productid=1) WHERE orderid = 1;

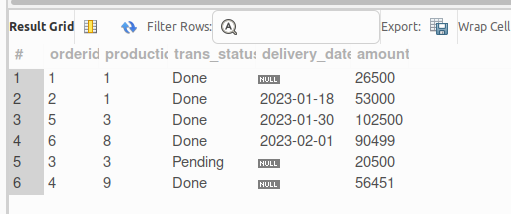
UPDATE order\_details SET amount = (SELECT p.price\*o.quantity FROM `order` o , `product` p WHERE o.orderid=2 AND p.productid=1) WHERE orderid = 2;

UPDATE order\_details SET amount = (SELECT p.price\*o.quantity FROM `order` o , `product` p WHERE o.orderid=5 AND p.productid=3) WHERE orderid = 5;

UPDATE order\_details SET amount = (SELECT p.price\*o.quantity FROM `order` o , `product` p WHERE o.orderid=6 AND p.productid=8) WHERE orderid = 6;

UPDATE order\_details SET amount = (SELECT p.price\*o.quantity FROM `order` o , `product` p WHERE o.orderid=3 AND p.productid=3) WHERE orderid = 3;

UPDATE order\_details SET amount = (SELECT p.price\*o.quantity FROM `order` o , `product` p WHERE o.orderid=4 AND p.productid=9) WHERE orderid = 4;



* **MySQL PRACTICAL QUERY:**

**1) Fetch all the User order list and include atleast following details in that.**

**- Customer name**

**- Product names**

**- Order Date**

**- Expected delivery date (in days, i.e. within X days)**

SELECT u.name AS "Customer Name", p.product\_name AS "Product Name", o.order\_date AS "Order Date",

CONCAT(DATEDIFF(o.expected\_delivery,o.order\_date)," days") AS "Expected Delivery"

FROM `order\_details` od

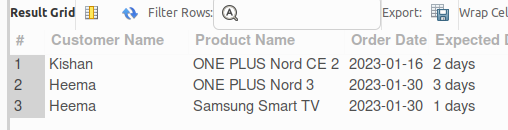
LEFT JOIN `order` o USING(`orderid`)

LEFT JOIN `product` p USING(productid)

LEFT JOIN `user` u USING(userid)

WHERE delivery\_date IS NOT NULL

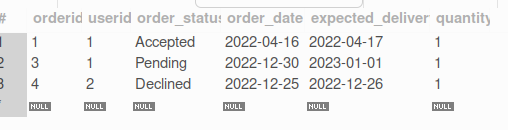
ORDER BY o.order\_date;



**2. Create summary report which provide information about**

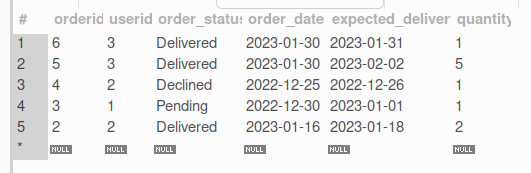
**- All undelivered Orders**

SELECT \* FROM `order` WHERE NOT order\_status = "Delivered";



**- 5 Most recent orders**

SELECT \* FROM `order` ORDER BY orderid DESC LIMIT 5;



**- Top 5 active users (Users having most number of orders)**

**OPTION 1:**

SELECT u.name AS "Active Users", count(o.quantity)

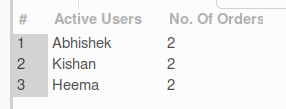
AS "No. Of Orders"

FROM `order` o

INNER JOIN `user` u USING(userid)

GROUP BY u.name

ORDER BY count(o.quantity) DESC LIMIT 5;



**- Inactive users (Users who hasn’t done any order)**

SELECT usr.name AS "Inactive Users" FROM `user` usr

WHERE usr.userid NOT IN (

SELECT DISTINCT o.userid

FROM `order` o INNER JOIN `user` u USING(userid)

);



**- Top 5 Most purchased products**

SELECT p.product\_name AS "Most Purchased Products", count(o.quantity) AS "Total Quantity"

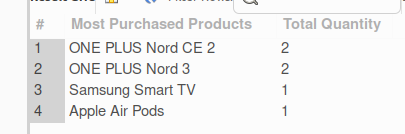
FROM `order\_details` od

INNER JOIN `product` p USING(productid)

INNER JOIN `order` o USING(orderid)

GROUP BY p.product\_name

ORDER BY count(o.quantity) DESC LIMIT 5;



**- Most expensive and most chepest orders.**

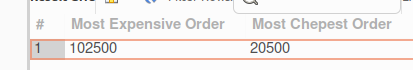
SELECT max(output.total) AS "Most Expensive Order", min(output.total) AS "Most Chepest Order" FROM (

SELECT p.price\*o.quantity AS total FROM `order\_details` od

INNER JOIN `order` o USING(orderid)

INNER JOIN `product` p USING(productid)

) AS output;



**OPTION 2:**

WITH main AS (SELECT p.product\_name AS pname, p.price as price, o.quantity as quantity

FROM `order\_details` od

INNER JOIN `order` o USING(orderid)

INNER JOIN `product` p USING(productid))

SELECT @total as "Total Amount", main.pname, "Most Chepest" AS "Order Status"

FROM main WHERE (@total:= main.price\*main.quantity) = (

SELECT MIN(main.price\*main.quantity) AS min

FROM main)

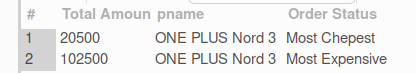
UNION

SELECT @total as "Total Amount", main.pname, "Most Expensive" AS "Order Status"

FROM main WHERE (@total:= main.price\*main.quantity) = (

SELECT MAX(main.price\*main.quantity) AS max

FROM main);



**OPTION 3:**

SELECT p.product\_name as "Product", od.amount as "AMOUNT", "Most Expensive" AS "Status"

FROM `order\_details` od INNER JOIN product p USING(productid) WHERE od.amount = (

SELECT MAX(odr.amount) FROM `order\_details` odr)

UNION

SELECT p.product\_name as "Product", od.amount as "AMOUNT", "Most Chepest" AS "Status"

FROM `order\_details` od INNER JOIN product p USING(productid) WHERE od.amount = (

SELECT MIN(odr.amount) FROM `order\_details` odr);

