MCA 2021-23 TRIM 4

Open Source Database Project

Project Report on

Product Inventory Management System(Mongodb)

Event Management System(Postgresql)

Presented By

38 - Parmeet Singh Sandhu

39 - Kunal Sarpe

40 - Saumik Dutta

41 - Pallavi Sawant

# Table of Contents

| Sr no | Title |
| --- | --- |
| 1 | Product Inventory Management System |
| a | Introduction |
| b | Queries |
|  |  |
| 2 | Event Management System |
| a | Introduction |
| b | Entity Relationship Diagram |
| c | ER Model to Relational Model |
| d | Queries |

**Product Inventory Management System**

Introduction

Product inventory management system is a management system by which you can track your goods throughout your entire supply chain, from purchasing to production to end sales. This system has the capacity to store many different types of objects with different sets of attributes.

Customers in e-commerce stores regularly add and remove items from their “shopping cart,” change quantities multiple times, abandon the cart at any point, and sometimes have problems during and after checkout that require a hold or canceled order. These activities make it difficult to maintain inventory systems and counts and ensure that customers cannot “buy” items that are unavailable while they shop in your store.

This management system keeps the traditional metaphor of the shopping cart.

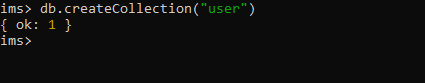
Queries

Creating database of inventory management system:



Creating collection ‘User’ to store user details:

db.createCollection("user")



Inserting records into ‘User’ collection:

db.user.insert({

... id:1,

... roleid:"Admin",

... firstname:"Kunal",

... middlename:"Pramod",

... lastname:"Sarpe",

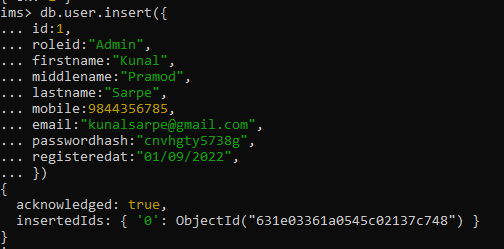
... mobile:9844356785,

... email:"kunalsarpe@gmail.com",

... passwordhash:"cnvhgty5738g",

... registeredat:"01/09/2022",

... })



db.user.insertMany({

... id:2,

... roleid:"Customer",

... firstname:"Dixit",

... middlename:"Harish",

... lastname:"Salian",

... mobile:8984475748,

... email:"dixitsalian@gmail.com",

... passwordhash:"cn6shdg36784",

... registeredat:"02/09/2022"

... },

... {

..... id:3,

..... roleid:"Customer",

..... firstname:"Parth",

..... middlename:"Pramod",

..... lastname:"Vesvikar",

..... mobile:7784673212,

..... email:"parthvesvikar@gmail.com",

..... passwordhash:"c746yhgurty7",

..... registeredat:"03/09/2022"

..... },

... {

..... id:4,

..... roleid:"Customer",

..... firstname:"Omkar",

..... middlename:"Sudhir",

..... lastname:"Parte",

..... mobile:9811234589,

..... email:"omkarparte@gmail.com",

..... passwordhash:"cvhfyr88uyrt",

..... registeredat:"04/09/2022"

..... },

... {

..... id:5,

..... roleid:"Customer",

..... firstname:"Ishpreet",

..... middlename:"Sodhi",

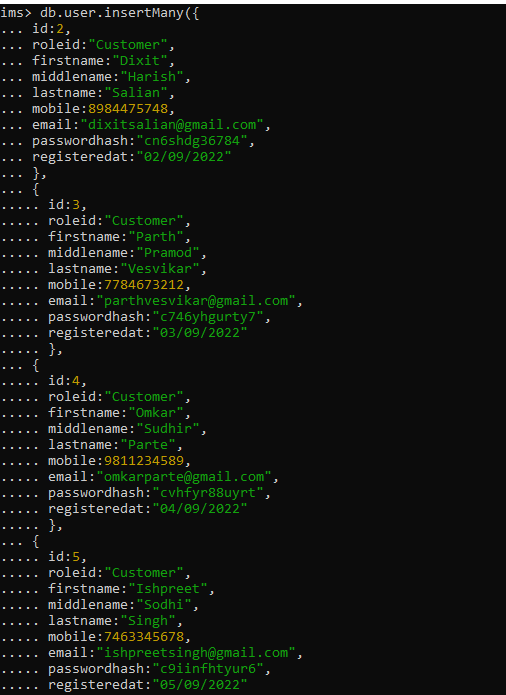
..... lastname:"Singh",

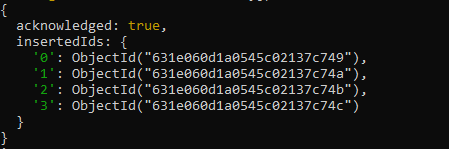
..... mobile:7463345678,

..... email:"ishpreetsingh@gmail.com",

..... passwordhash:"c9iinfhtyur6",

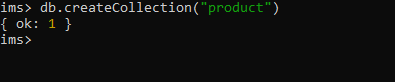
..... registeredat:"05/09/2022"





Creating collection “product” to store product details:

db.createCollection("product")



Inserting records into “product” collection:

db.product.insert({

... id:3,

... title:"Nintendo Switch OLED Splatoon 3 Special Edition",

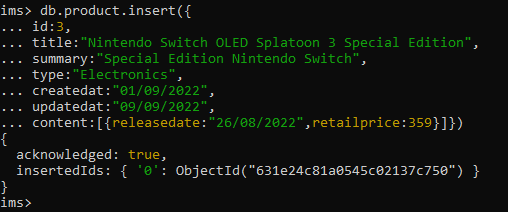
... summary:"Special Edition Nintendo Switch",

... type:"Electronics",

... createdat:"01/09/2022",

... updatedat:"09/09/2022",

... content:[{releasedate:"26/08/2022",retailprice:359}]})



db.product.insert({

... id:4,

... title:"2021 Panini Select Football Hobby Box",

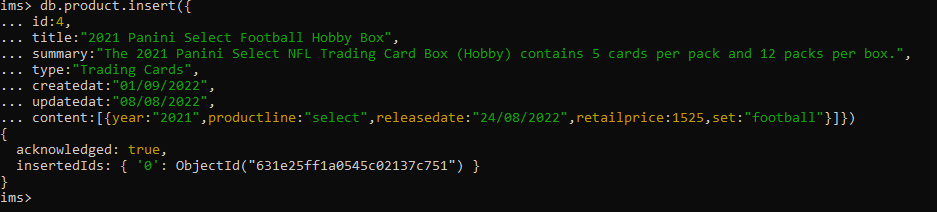
... summary:"The 2021 Panini Select NFL Trading Card Box (Hobby) contains 5 cards per pack and 12 packs per box.",

... type:"Trading Cards",

... createdat:"01/09/2022",

... updatedat:"08/08/2022",

... content:[{year:"2021",productline:"select",releasedate:"24/08/2022",retailprice:1525,set:"football"}]})



db.product.insert({

... id:5,

... title:"Supreme ty Beanie Baby Multicolor",

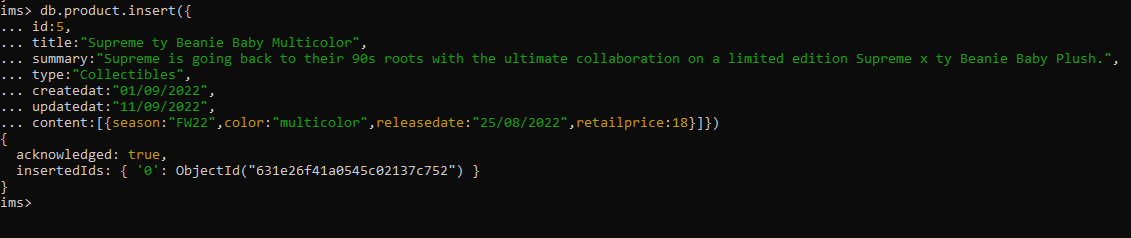
... summary:"Supreme is going back to their 90s roots with the ultimate collaboration on a limited edition Supreme x ty Beanie Baby Plush.",

... type:"Collectibles",

... createdat:"01/09/2022",

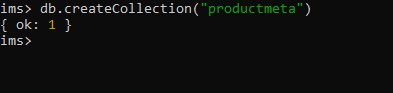
... updatedat:"11/09/2022",

... content:[{season:"FW22",color:"multicolor",releasedate:"25/08/2022",retailprice:18}]})



Creating collection “Product Meta” to store additional information about product:

db.createCollection("productmeta")



Inserting records in “Product Meta” collection:

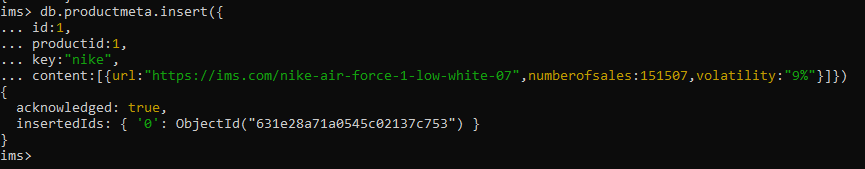
db.productmeta.insert({

... id:1,

... productid:1,

... key:"nike",

... content:[{url:"https://ims.com/nike-air-force-1-low-white-07",numberofsales:151507,volatility:"9%"}]})



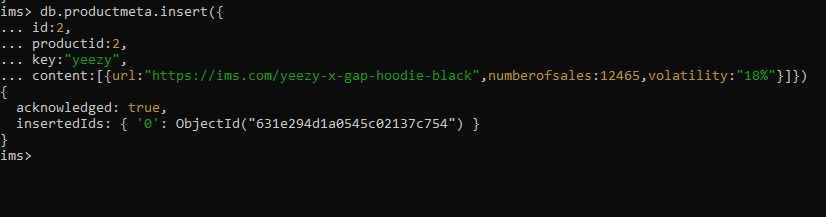
db.productmeta.insert({

... id:2,

... productid:2,

... key:"yeezy",

... content:[{url:"https://ims.com/yeezy-x-gap-hoodie-black",numberofsales:12465,volatility:"18%"}]})



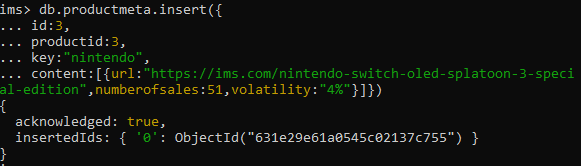
db.productmeta.insert({

... id:3,

... productid:3,

... key:"nintendo",

... content:[{url:"https://ims.com/nintendo-switch-oled-splatoon-3-special-edition",numberofsales:51,volatility:"4%"}]})



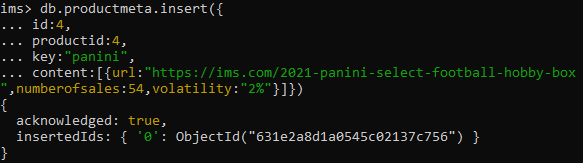
db.productmeta.insert({

... id:4,

... productid:4,

... key:"panini",

... content:[{url:"https://ims.com/2021-panini-select-football-hobby-box",numberofsales:54,volatility:"2%"}]})



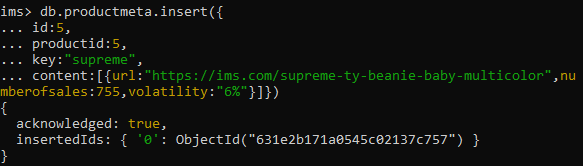
db.productmeta.insert({

... id:5,

... productid:5,

... key:"supreme",

... content:[{url:"https://ims.com/supreme-ty-beanie-baby-multicolor",numberofsales:755,volatility:"6%"}]})



Creating collection “category” to categorize products:

db.createCollection("category")



Inserting records in “category” collection:

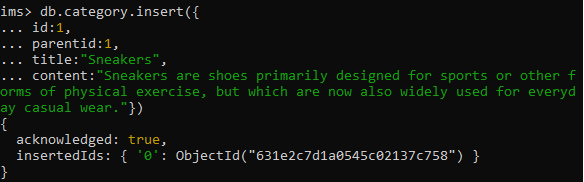
db.category.insert({

... id:1,

... parentid:1,

... title:"Sneakers",

... content:"Sneakers are shoes primarily designed for sports or other forms of physical exercise, but which are now also widely used for everyday casual wear."})



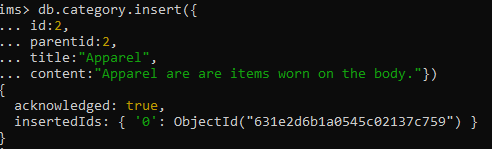
db.category.insert({

... id:2,

... parentid:2,

... title:"Apparel",

... content:"Apparel are are items worn on the body."})



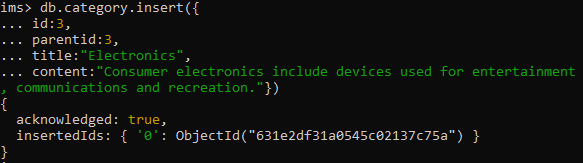
db.category.insert({

... id:3,

... parentid:3,

... title:"Electronics",

... content:"Consumer electronics include devices used for entertainment, communications and recreation."})



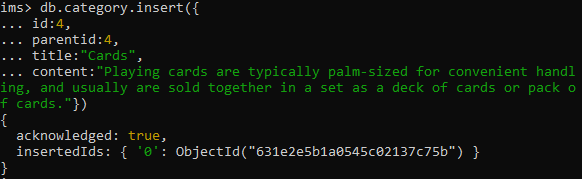
db.category.insert({

... id:4,

... parentid:4,

... title:"Cards",

... content:"Playing cards are typically palm-sized for convenient handling, and usually are sold together in a set as a deck of cards or pack of cards."})



Creating collection “product category” to map products to categories:

db.createCollection("productcategory")

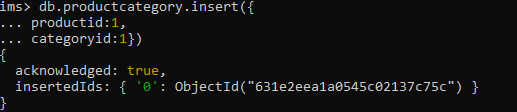


Inserting records in collection “productcategory”:

db.productcategory.insert({

... productid:1,

... categoryid:1})

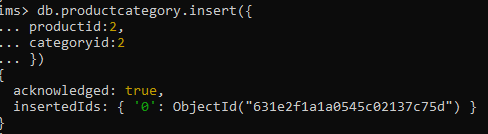


db.productcategory.insert({

... productid:2,

... categoryid:2

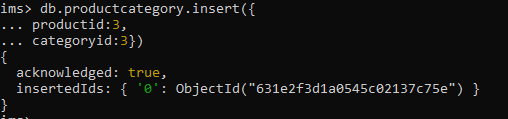
... })



db.productcategory.insert({

... productid:3,

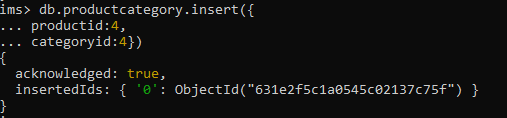
... categoryid:3})



db.productcategory.insert({

... productid:4,

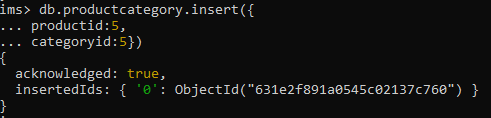
... categoryid:4})



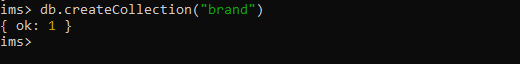
db.productcategory.insert({

... productid:5,

... categoryid:5})



Creating collection “brand” to store brand data:



Inserting records into “brand” collection:

db.brand.insert({

... id:1,

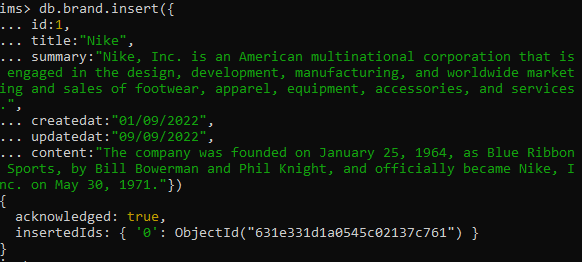
... title:"Nike",

... summary:"Nike, Inc. is an American multinational corporation that is engaged in the design, development, manufacturing, and worldwide marketing and sales of footwear, apparel, equipment, accessories, and services.",

... createdat:"01/09/2022",

... updatedat:"09/09/2022",

... content:"The company was founded on January 25, 1964, as "Blue Ribbon Sports", by Bill Bowerman and Phil Knight, and officially became Nike, Inc. on May 30, 1971."})



db.brand.insert({

... id:2,

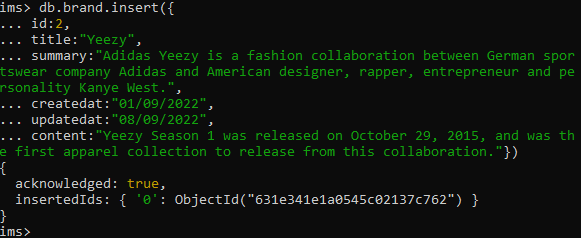
... title:"Yeezy",

... summary:"Adidas Yeezy is a fashion collaboration between German sportswear company Adidas and American designer, rapper, entrepreneur and personality Kanye West.",

... createdat:"01/09/2022",

... updatedat:"08/09/2022",

... content:"Yeezy Season 1 was released on October 29, 2015, and was the first apparel collection to release from this collaboration."})



db.brand.insert({

... id:3,

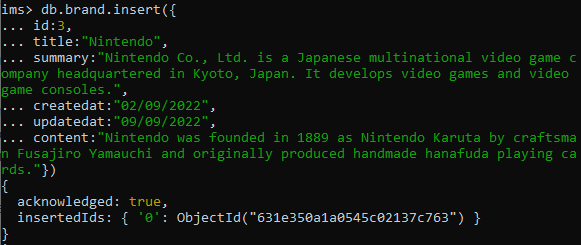
... title:"Nintendo",

... summary:"Nintendo Co., Ltd. is a Japanese multinational video game company headquartered in Kyoto, Japan. It develops video games and video game consoles.",

... createdat:"02/09/2022",

... updatedat:"09/09/2022",

... content:"Nintendo was founded in 1889 as Nintendo Karuta by craftsman Fusajiro Yamauchi and originally produced handmade hanafuda playing cards."})



db.brand.insert({

... id:4,

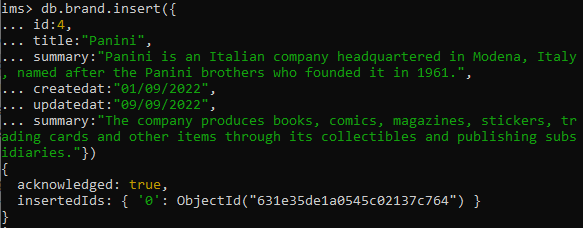
... title:"Panini",

... summary:"Panini is an Italian company headquartered in Modena, Italy, named after the Panini brothers who founded it in 1961.",

... createdat:"01/09/2022",

... updatedat:"09/09/2022",

... summary:"The company produces books, comics, magazines, stickers, trading cards and other items through its collectibles and publishing subsidiaries."})



db.brand.insert({

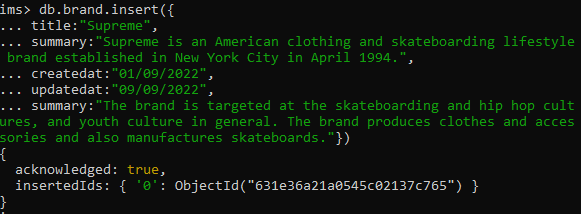
... title:"Supreme",

... summary:"Supreme is an American clothing and skateboarding lifestyle brand established in New York City in April 1994.",

... createdat:"01/09/2022",

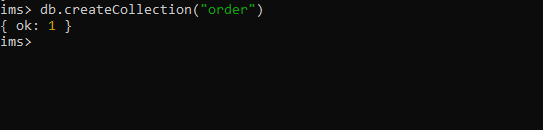
... updatedat:"09/09/2022",

... summary:"The brand is targeted at the skateboarding and hip hop cultures, and youth culture in general. The brand produces clothes and accessories and also manufactures skateboards."})



Creating collection “order” to manage the inventory orders. The order can be associated with either Supplier or the Customer.

db.createCollection("order")



Inserting records into the “order” collection:

db.order.insert({

... id:1,

... userid:2,

... type:2,

... status:"New",

... subtotal:230.00,

... itemdiscount:10,

... tax:12,

... shipping:30,

... total:272,

... promo:"HALLOWEEN",

... discount:20,

... grandtotal:217.6,

... createdat:"10/09/2022",

... updatedat:"11/09/2022",

... content:"Apparel"})



db.order.insert({

... id:2,

... userid:1,

... type:1,

... status:"Paid",

... subtotal:1000.00,

... itemdiscount:10,

... tax:50,

... shipping:100,

... total:1150.00,

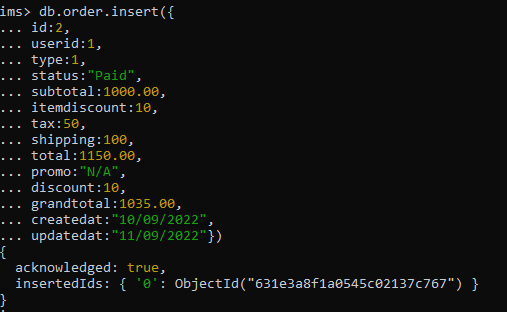
... promo:"N/A",

... discount:10,

... grandtotal:1035.00,

... createdat:"10/09/2022",

... updatedat:"11/09/2022"})



db.order.insert({

... id:3,

... userid:2,

... type:2,

... status:"Checkout",

... subtotal:125.00,

... itemdiscount:5,

... tax:10,

... shipping:30,

... total:165.00,

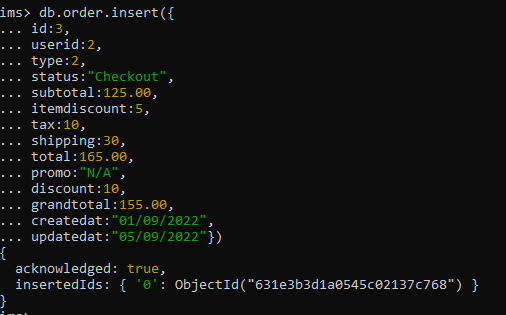
... promo:"N/A",

... discount:10,

... grandtotal:155.00,

... createdat:"01/09/2022",

... updatedat:"05/09/2022"})



db.order.insert({

... id:4,

... userid:2,

... type:2,

... status:"Failed",

... subtotal:455.00,

... itemdiscount:20,

... tax:16.77,

... shipping:50,

... total:521.77,

... promo:"XMAS2022",

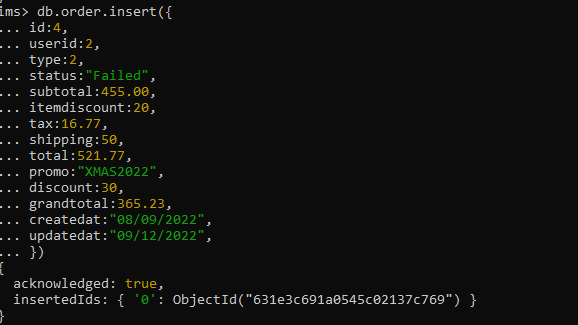
... discount:30,

... grandtotal:365.23,

... createdat:"08/09/2022",

... updatedat:"09/12/2022",

... })



db.order.insert({

... id:5,

... userid:1,

... type:1,

... status:"Shipped",

... subtotal:1500.00,

... itemdiscount:10,

... tax:65.23,

... shipping:200,

... total:1765.23,

... promo:"N/A",

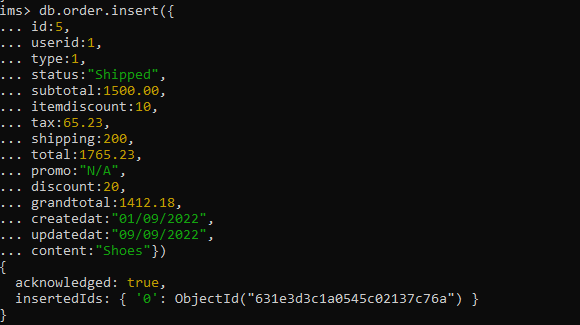
... discount:20,

... grandtotal:1412.18,

... createdat:"01/09/2022",

... updatedat:"09/09/2022",

... content:"Shoes"})



Creating collection “address” to manage the address of either the user or the order. The user address can be used to store the address associated with the user. The order address can be used to store the delivery address for the home delivery orders.

db.createCollection("address")



Inserting records into “address” collection:

db.address.insert({

... id:1,

... userid:1,

... orderid:1,

... firstname:"Kunal",

... middlename:"Pramod",

... lastname:"Sarpe",

... mobile:9844356785,

... email:"kunalsarpe@gmail.com",

... line1:"4024 Marietta Street",

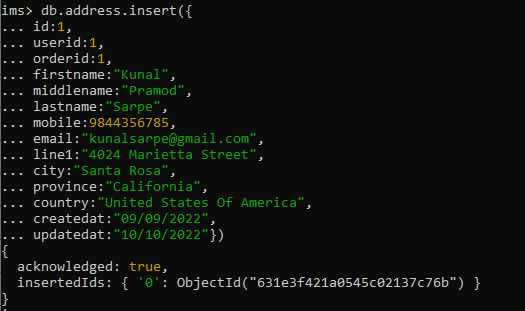
... city:"Santa Rosa",

... province:"California",

... country:"United States Of America",

... createdat:"09/09/2022",

... updatedat:"10/10/2022"})



db.address.insert({

... id:2,

... userid:2,

... orderid:2,

... firstname:"Dixit",

... middlename:"Harish",

... lastname:"Salian",

... mobile:8984475748,

... email:"dixitsalian@gmail.com",

... line1:"3929 Griffin Street",

... city:"Phoenix",

... province:"Arizona",

... country:"United States Of America",

... createdat:"01/09/2022",

... updatedat:"09/09/2022"})



db.address.insert({

... id:3,

... userid:3,

... orderid:3,

... firstname:"Parth",

... middlename:"Pramod",

... lastname:"Vesvikar",

... mobile:7784673212,

... email:"parthvesvikar@gmail.com",

... line1:"501 Oak Ridge Drive",

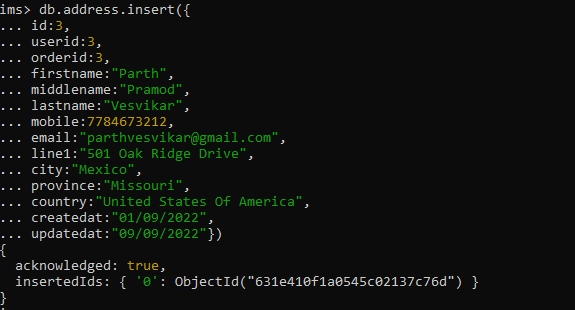
... city:"Mexico",

... province:"Missouri",

... country:"United States Of America",

... createdat:"01/09/2022",

... updatedat:"09/09/2022"})



db.address.insert({

... id:4,

... userid:4,

... orderid:4,

... firstname:"Omkar",

... middlename:"Sudhir",

... lastname:"Parte",

... mobile:9811234589,

... email:"omkarparte@gmail.com",

... line1:"4552 Colonial Drive",

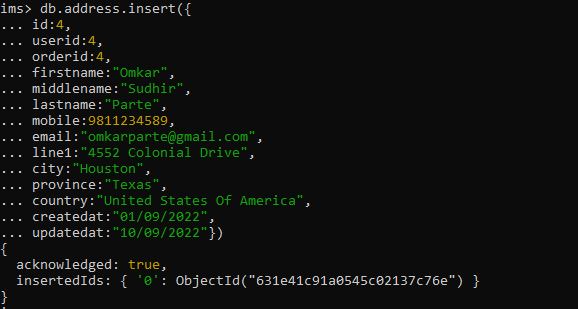
... city:"Houston",

... province:"Texas",

... country:"United States Of America",

... createdat:"01/09/2022",

... updatedat:"10/09/2022"})



db.address.insert({

... id:5,

... userid:5,

... orderid:5,

... firstname:"Ishpreet",

... middlename:"Sodhi",

... lastname:"Singh",

... mobile:7463345678,

... email:"ishpreetsingh@gmail.com",

... line1:"3691 Shady Pines Drive",

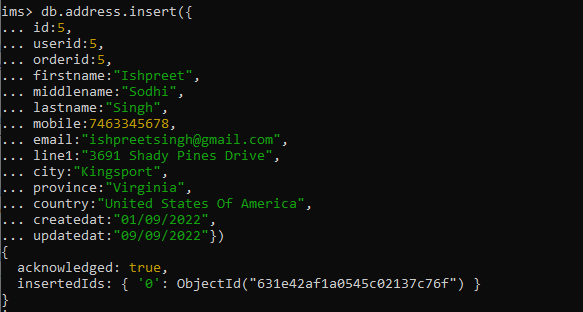
... city:"Kingsport",

... province:"Virginia",

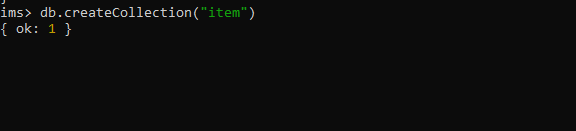
... country:"United States Of America",

... createdat:"01/09/2022",

... updatedat:"09/09/2022"})



Creating a collection “item” to store the item details. The Item represents the items stocked in the inventory and purchased from the suppliers.



Inserting records into collection “item”:

db.item.insert({

... id:1,

... productid:1,

... brandid:1,

... supplierid:10001,

... orderid:5,

... createdby:1,

... updatedby:1,

... sku:10001,

... mrp:110,

... discount:10,

... price:60,

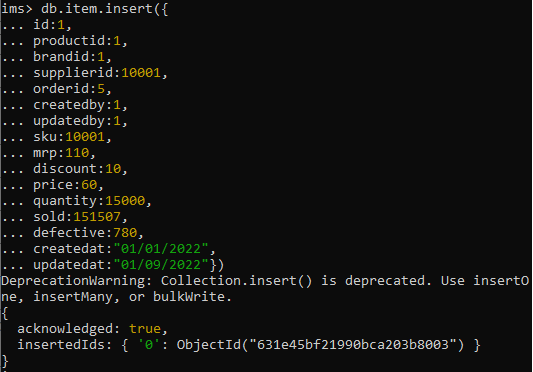
... quantity:15000,

... sold:151507,

... defective:780,

... createdat:"01/01/2022",

... updatedat:"01/09/2022"})



db.item.insert({

... id:2,

... productid:2,

... brandid:2,

... supplierid:10002,

... orderid:2,

... createdby:1,

... updatedby:1,

... sku:10002,

... mrp:90,

... discount:0,

... price:70,

... quantity:670,

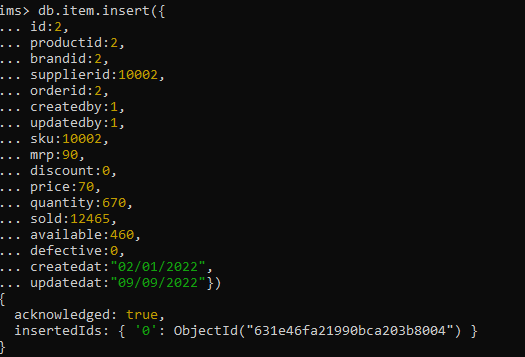
... sold:12465,

... available:460,

... defective:0,

... createdat:"02/01/2022",

... updatedat:"09/09/2022"})



db.item.insert({

... id:3,

... productid:3,

... brandid:3,

... supplierid:10003,

... orderid:6,

... createdby:6,

... updatedby:7,

... sku:10003,

... mrp:359,

... discount:10,

... price:323.1,

... quantity:45000,

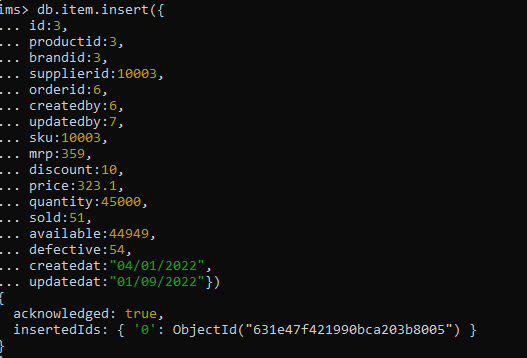
... sold:51,

... available:44949,

... defective:54,

... createdat:"04/01/2022",

... updatedat:"01/09/2022"})



db.item.insert({

... id:4,

... productid:4,

... brandid:4,

... supplierid:10007,

... orderid:9,

... createdby:11,

... updatedby:7,

... sku:10004,

... mrp:1525,

... discount:10,

... price:1372.5,

... quantity:5000,

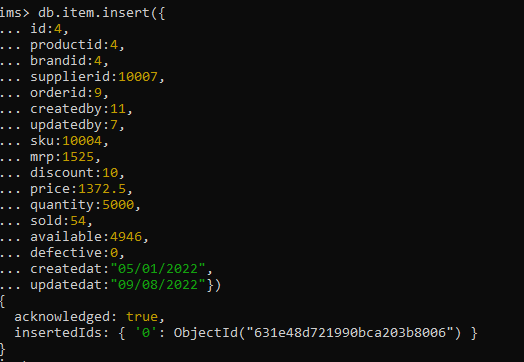
... sold:54,

... available:4946,

... defective:0,

... createdat:"05/01/2022",

... updatedat:"09/08/2022"})



db.item.insert({

... id:5,

... productid:5,

... brandid:5,

... supplierid:10010,

... orderid:11,

... createdby:15,

... updatedby:7,

... sku:10005,

... mrp:18,

... discount:0,

... price:18,

... quantity:125,

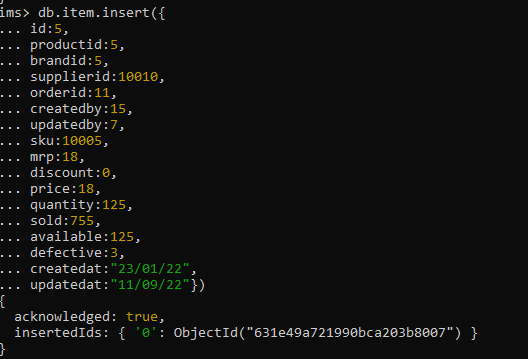
... sold:755,

... available:125,

... defective:3,

... createdat:"23/01/22",

... updatedat:"11/09/22"})



Create collection “orderitem” to manage the order items purchased by the customers.

db.createCollection("orderitem")



db.orderitem.insert({

... id:1,

... productid:1,

... itemid:1,

... orderid:1,

... sku:10001,

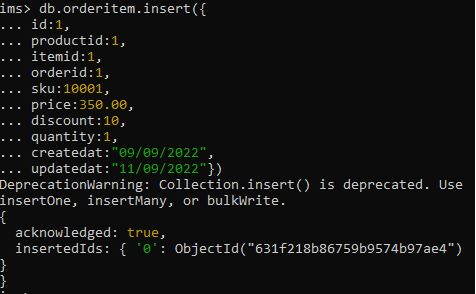
... price:350.00,

... discount:10,

... quantity:1,

... createdat:"09/09/2022",

... updatedat:"11/09/2022"})



db.orderitem.insertMany([

... {

..... id:2,

..... productid:2,

..... itemid:2,

..... orderid:2,

..... sku:10002,

..... price:212.00,

..... discount:0,

..... quantity:2,

..... createdat:"09/09/2022",

..... updatedat:"12/09/2022"},

... {

..... id:3,

..... productid:3,

..... itemid:3,

..... orderid:3,

..... sku:10003,

..... price:100.00,

..... discount:5,

..... quantity:2,

..... createdat:"08/09/2022",

..... updatedat:"10/09/2022"},

... {

..... id:4,

..... productid:4,

..... itemid:4,

..... orderid:4,

..... sku:10004,

..... price:200.00,

..... discount:10,

..... quantity:1,

..... createdat:"05/09/2022",

..... updatedat:"06/09/2022"},

... {

..... id:5,

..... productid:5,

..... itemid:5,

..... orderid:5,

..... sku:10005,

..... price:1200,

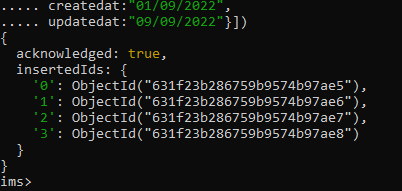
..... discount:5,

..... quantity:1,

..... createdat:"01/09/2022",

..... updatedat:"09/09/2022"}])





Creating a collection “transaction” to track the order payments made by the buyer and for bookkeeping. We can also use the same table to record the partial or full refund of the order.

db.createCollection("transaction")





Inserting records into “transaction”:

db.transaction.insert({

... id:1,

... userid:1,

... orderid:1,

... code:"axfhryu321bh",

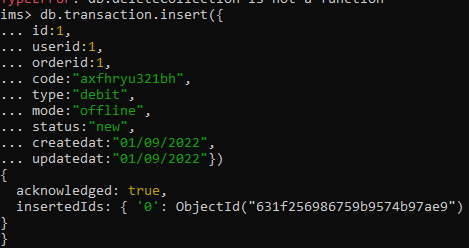
... type:"debit",

... mode:"offline",

... status:"new",

... createdat:"01/09/2022",

... updatedat:"01/09/2022"})



db.transaction.insert({

... id:2,

... userid:2,

... orderid:2,

... code:"yuioe4r5tu89",

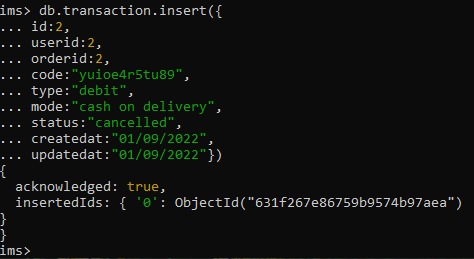
... type:"debit",

... mode:"cash on delivery",

... status:"cancelled",

... createdat:"01/09/2022",

... updatedat:"01/09/2022"})



db.transaction.insert({

... id:3,

... userid:3,

... orderid:3,

... code:"mnvbcgdbgh12",

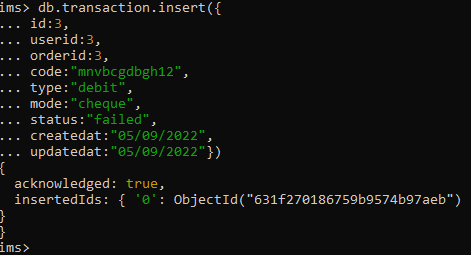
... type:"debit",

... mode:"cheque",

... status:"failed",

... createdat:"05/09/2022",

... updatedat:"05/09/2022"})



db.transaction.insert({

... id:4,

... userid:4,

... orderid:4,

... code:"xbcvetw83gd5",

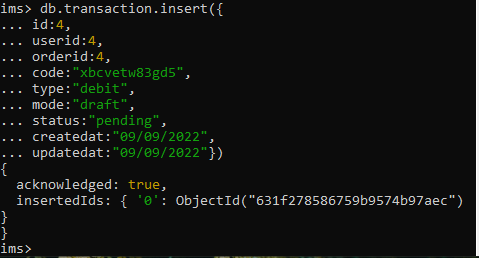
... type:"debit",

... mode:"draft",

... status:"pending",

... createdat:"09/09/2022",

... updatedat:"09/09/2022"})



db.transaction.insert({

... id:5,

... userid:5,

... orderid:5,

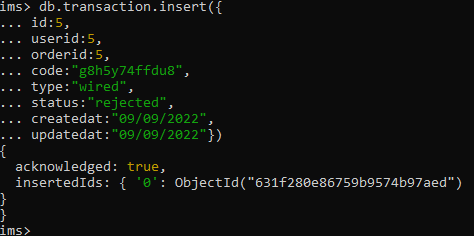
... code:"g8h5y74ffdu8",

... type:"wired",

... status:"rejected",

... createdat:"09/09/2022",

... updatedat:"09/09/2022"})



db.transaction.insert({

... id:6,

... userid:6,

... orderid:6,

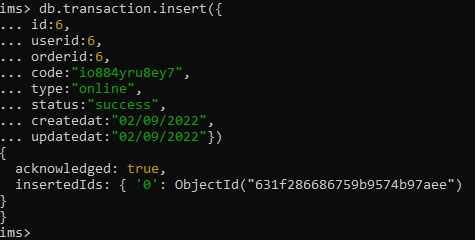
... code:"io884yru8ey7",

... type:"online",

... status:"success",

... createdat:"02/09/2022",

... updatedat:"02/09/2022"})



Queries:

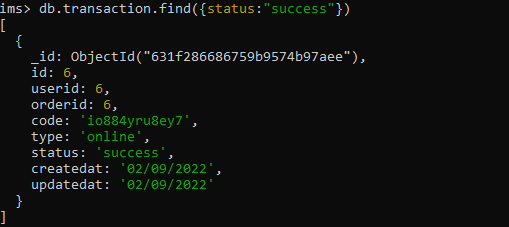
1] Field selection:

db.item.find()



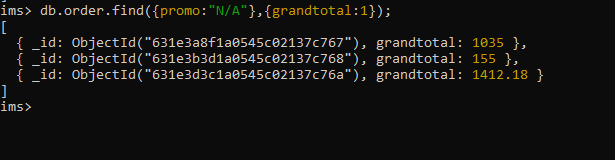
2] Condition clause: Where clause:

db.transaction.find({status:"success"})



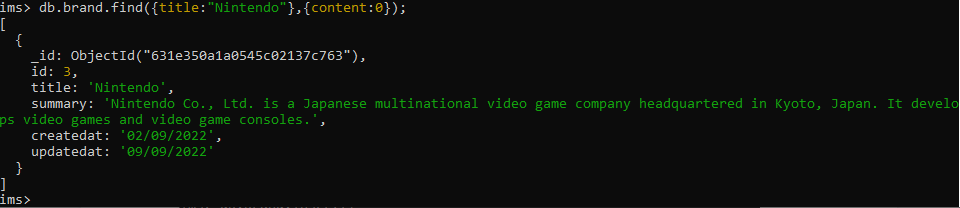
3] Projection: Select specific fields:

db.order.find({promo:"N/A"},{grandtotal:1});



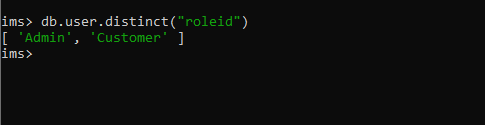
4] Field exclusion:

db.brand.find({title:"Nintendo"},{content:0});



5] Get distinct status:

db.user.distinct("roleid")



6] Sort:

db.item.find().sort({defective:1})





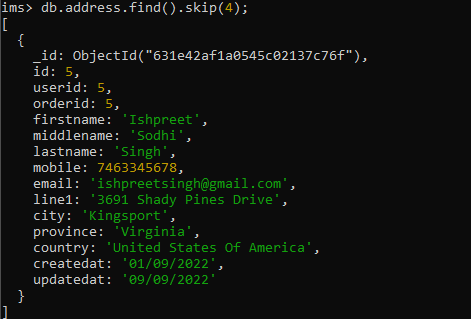
7] Limit:

db.order.find().limit(2);



8] Skip:

db.address.find().skip(4);



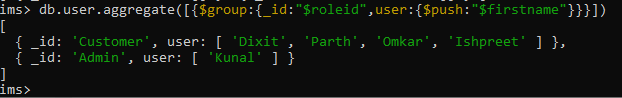
9] Aggregate:

db.orderitem.find().count();



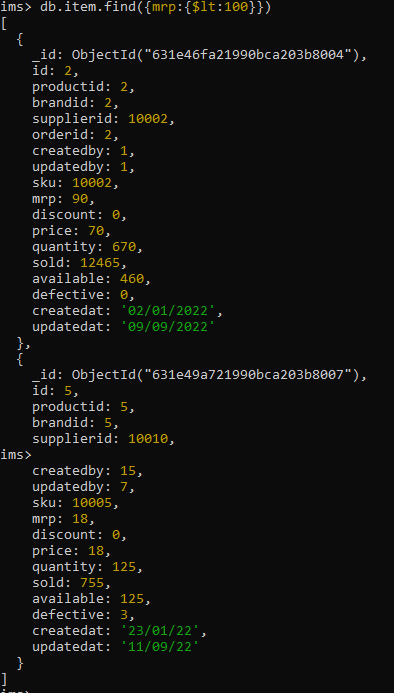
10] Group:

db.user.aggregate([{$group:{\_id:"$roleid",user:{$push:"$firstname"}}}])



11] Comparison:

db.item.find({mrp:{$lt:100}})



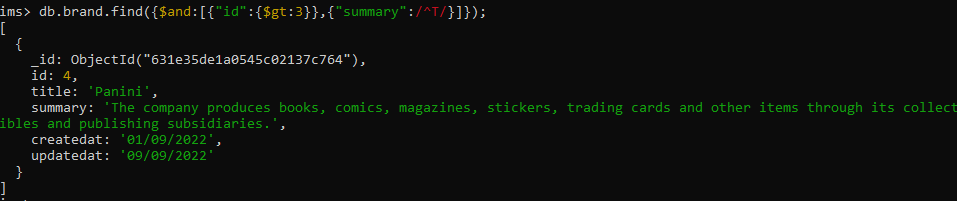
12] The following example query includes the $in operator, and will return documents whose city value matches either Mexico or Houston:

db.address.find({"city":{$in:["Mexico","Houston"]}})



13] The following example query includes the $and operator and fetches brand details with summary starting with “T” and id greater than 3:

db.brand.find({$and:[{"id":{$gt:3}},{"summary":/^T/}]});



14] The following example query includes the . operator to demonstrate nested query and fetches productmeta details with numberofsales greater than 10000:

db.productmeta.find({"content.numberofsales":{$gt:10000}})



15] The following example query shows the demonstration of indexes in MongoDB:

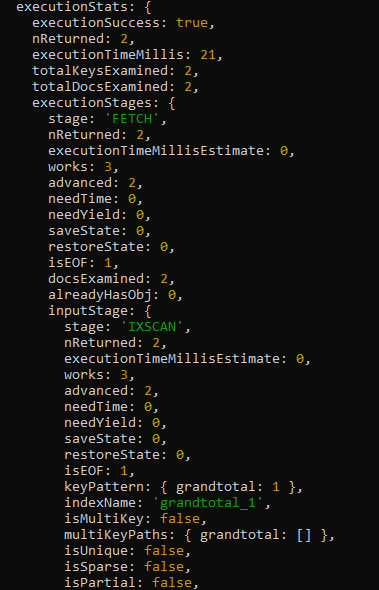
db.order.find({"grandtotal":{$gt:1000}}).explain("executionStats")



db.order.createIndex({"grandtotal":1})



db.order.find({"grandtotal":{$gt:1000}}).explain("executionStats")



# **Event Management System**

**Introduction:**

Event Management System is a System where a customer can approach a event organising company for hosting and managing events such as birthdays, award functions,wedding , reception , naming ceremony ,etc.

The requirement for automating the record keeping process of the event management system was to provide an efficient way of storing records and to provide a better alternative for the traditional way of storing records.

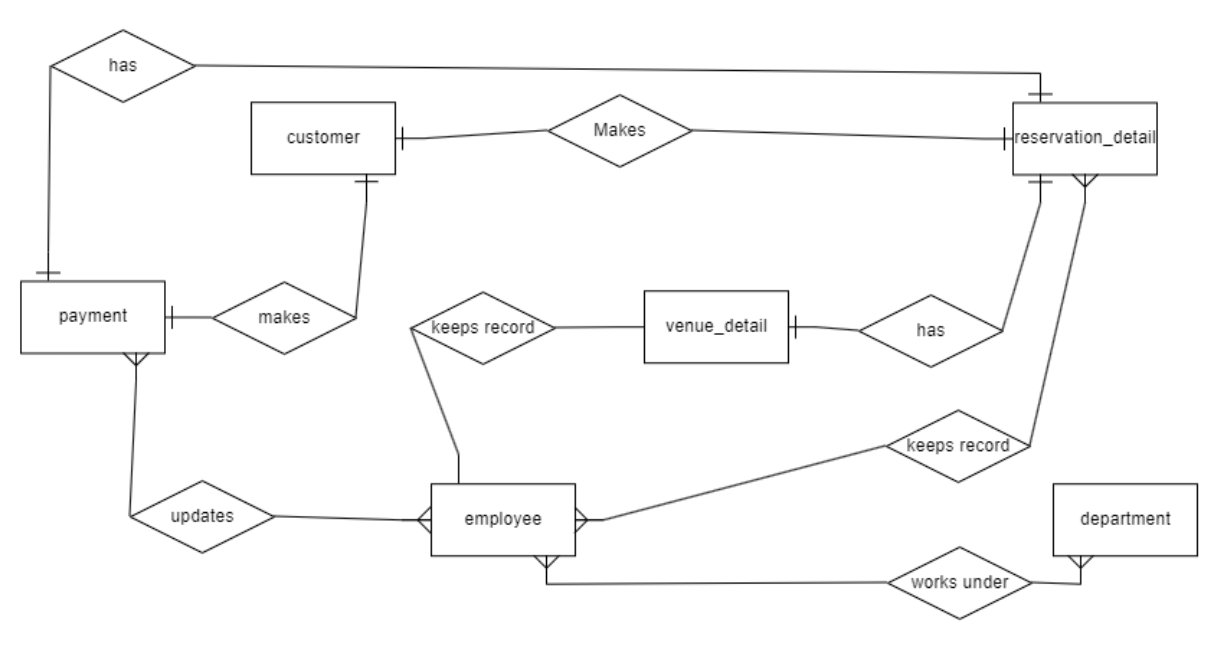
Event management System Database consist of the following tables

* Customer
* Employee
* Department
* Venue detail
* Reservation detail
* Payment

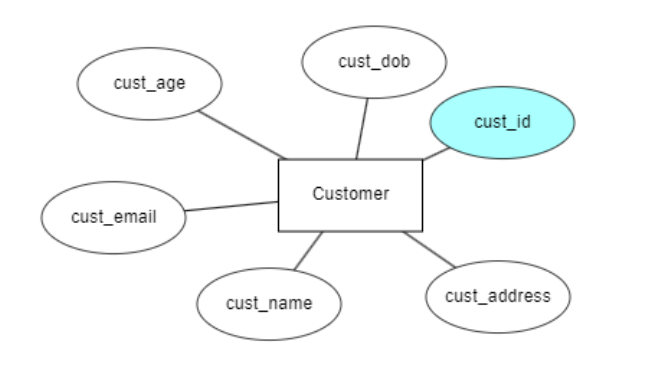
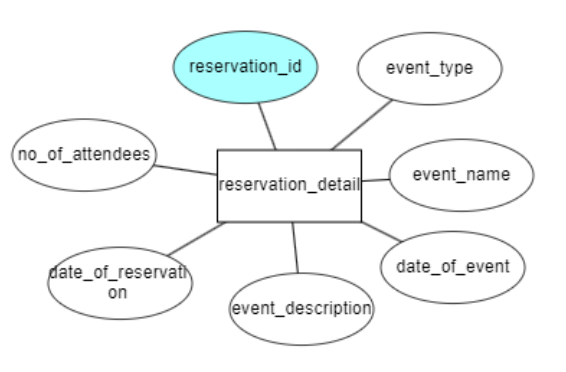
Below listed details are requirements of the system

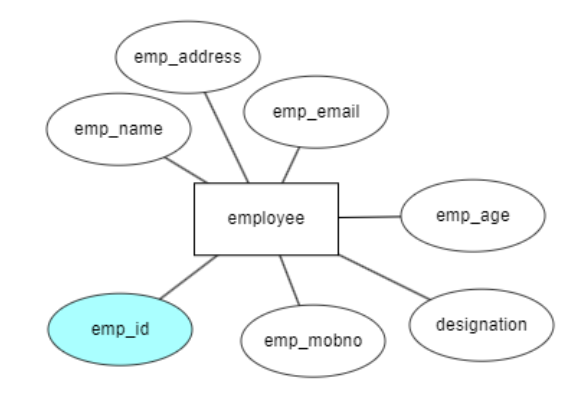
1. Operating system - Windows 8 and above.
2. Ram- 8 gb ddr4 and above
3. Storage - 3 GB
4. Software:- PostGres sql (SQL shell)

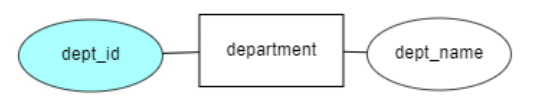
**Entity Relationship Diagram:**

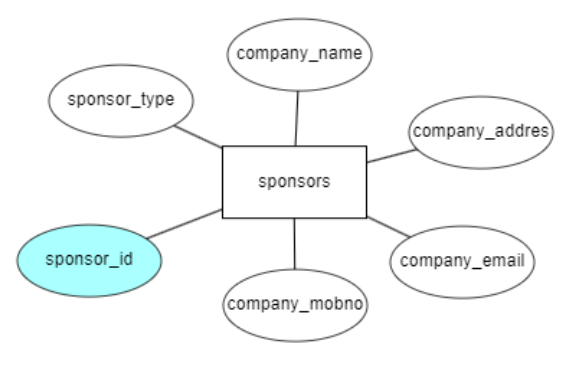


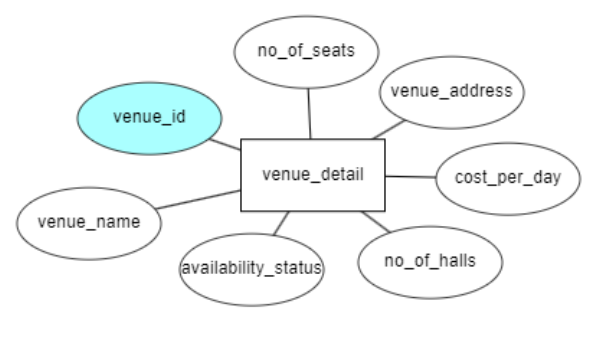
**ENTITIES:**

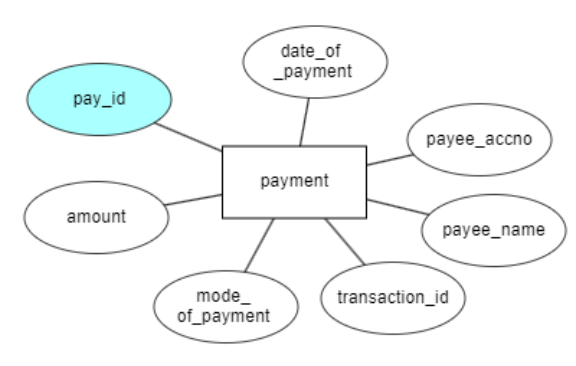
 





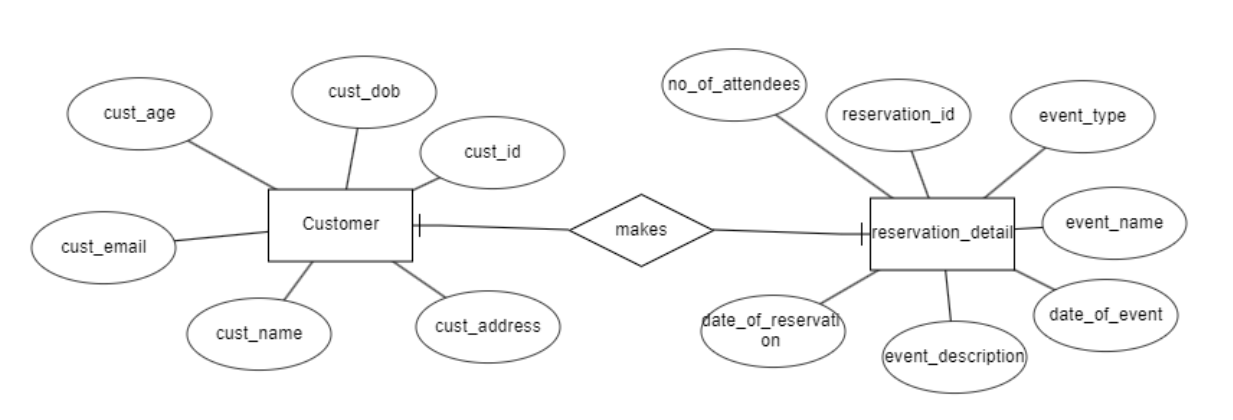






**Entity Relationship Diagram to Relational Model:**

1. Customer and Reservation:



Customer

| cust\_id | cust\_name | cust\_email | cust\_mobno | cust\_address | cust\_age | cust\_dob |
| --- | --- | --- | --- | --- | --- | --- |
| 1000 | Shobit sharma | shobit@gmail.com | 9874327569 | nerul | 27 | 12-05-1995 |

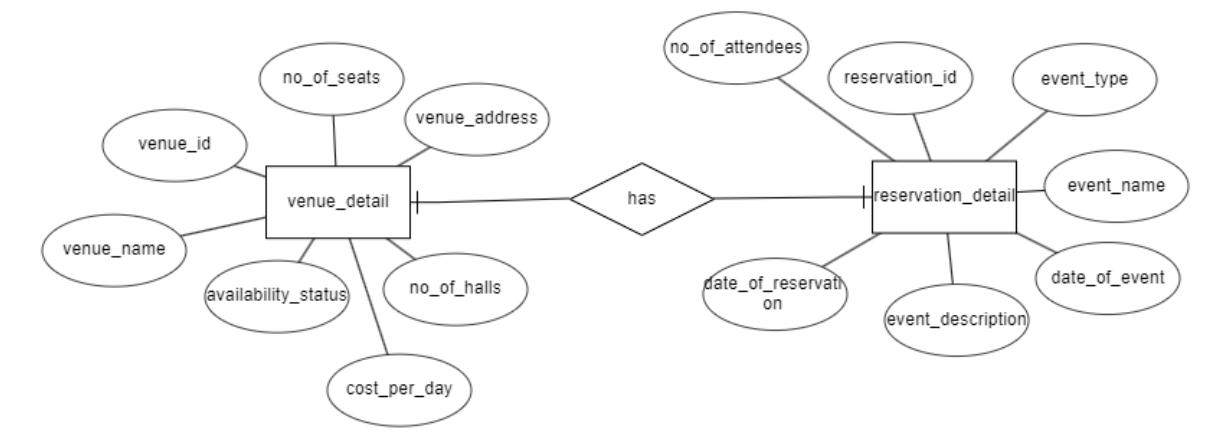
Reservation\_detail

| r\_id | cust\_id | event\_name | date\_of\_event | venue\_id | event\_desc | no\_of\_attendees | event\_type |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 5000 | 1000 | Birthday party' | 12-07-2021 | 4001 | Birthday party | 500 | offline |

Reason:-

Here customer id is foreign key to reservation detail as it helps in identifying the customer name who has done the reservation through cust\_id in reservation\_detail

2. Reservation and Venue:



Reservation\_detail

| r\_id | cust\_id | event\_name | date\_of\_event | venue\_id | event\_desc | no\_of\_attendees | event\_type |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 5000 | 1000 | Birthday party' | 12-07-2021 | 4001 | Birthday party | 500 | offline |

Venue\_detail

| venue\_id | venue\_name | no\_of\_seats | no\_of\_halls | cost\_per\_day | availability\_status | venue\_address |
| --- | --- | --- | --- | --- | --- | --- |
| 4001 | Four Points by Sheraton | 5000 | 5 | 12000 | avialable | Four Points by Sheraton vashi sec-27 |

Reason:-

Here venue id is foreign key to reservation detail as it helps in identifying the venue name where evnet shall be hosted through venue\_id in reservation\_detail

**Table Creation & Record Insert:**

**1. Customer details:**

create table customer(

cust\_id int ,

cust\_name character varying(100),

cust\_email varchar(50),

cust\_mobno varchar(15),

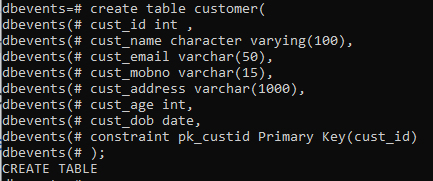
cust\_address varchar(1000),

cust\_age int,

cust\_dob date,

constraint pk\_custid Primary Key(cust\_id)

);

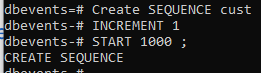


**Creating sequence for generating id:**

Create SEQUENCE cust

INCREMENT 1

START 1000 ;



**Inserting records in customer table:**

insert into customer(cust\_id,cust\_name,cust\_email,cust\_mobno,cust\_address,cust\_age,cust\_dob)

VALUES

(nextval('cust'),'Shobit sharma','shobit@gmail.com','9874327569','nerul',27,'1995-05-12'),

(nextval('cust'),'pooja dixit','pooja.d@gmail.com','9824327631','sanpada',26,'1996-07-21'),

(nextval('cust'),'Sidhant joshi','sid.j@gmail.com','9274327511','dadar',32,'1990-09-07'),

(nextval('cust'),'Anjali Mehta','anajali.m@gmail.com','9424327612',' andheri',40,'1983-07-21'),

(nextval('cust'),'Sakshi shetty','shakshi.s@gmail.com','8874327513',' parel',25,'1997-07-12');

insert into customer(cust\_id,cust\_name,cust\_email,cust\_mobno,cust\_address,cust\_age,cust\_dob)

VALUES

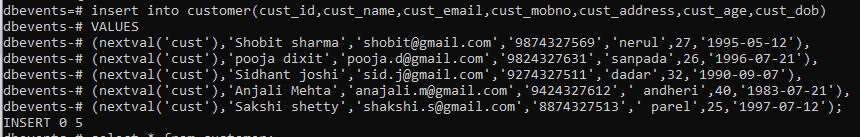
(nextval('cust'),'pallavi sawant','pallavi@gmail.com','8097809891','airoli',23,'2000-01-13'),

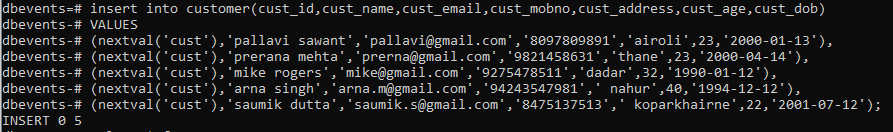
(nextval('cust'),'prerana mehta','prerna@gmail.com','9821458631','thane',23,'2000-04-14'),

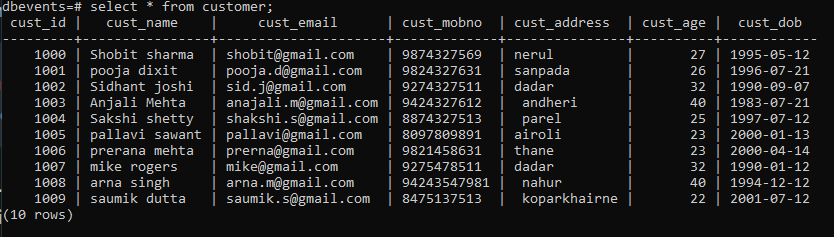
(nextval('cust'),'mike rogers','mike@gmail.com','9275478511','dadar',32,'1990-01-12'),

(nextval('cust'),'arna singh','arna.m@gmail.com','94243547981',' nahur',40,'1994-12-12'),

(nextval('cust'),'saumik dutta','saumik.s@gmail.com','8475137513',' koparkhairne',22,'2001-07-12');







**2. Employee:**

Create table employee(

empid int,

empname character varying(100),

emp\_age int,

emp\_mobno character varying(15),

emp\_email character varying(50),

emp\_address character varying(1000),

date\_of\_joining date,

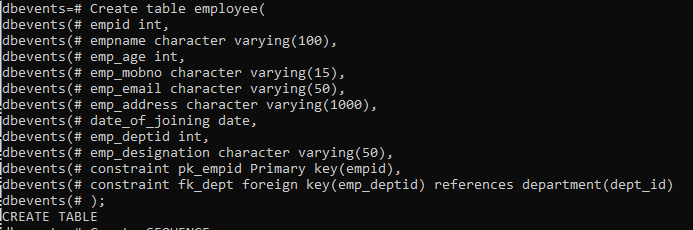
emp\_deptid int,

emp\_designation character varying(50),

constraint pk\_empid Primary key(empid),

constraint fk\_dept foreign key(emp\_deptid) references department(dept\_id)

);

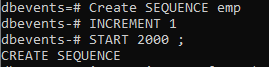


**Creating sequence for generating id:**

Create SEQUENCE emp

INCREMENT 1

START 2000 ;



**Inserting records in employee table;**

insert into employee(empid,empname,emp\_age,emp\_mobno,emp\_email,emp\_address,date\_of\_joining,emp\_deptid,emp\_designation)

values

(nextval('emp'),'sanjay sharma',27,'8874342211','sanjay.s@gmail.com','vashi','2018-04-01',3000,'social media manager'),

(nextval('emp'),'abhijeet Das',25,'8884342212','abhijeet.d@gmail.com','bhandup','2019-07-04',3001,'financial analyst'),

(nextval('emp'),'kanika khandelwal',23,'8872342213','kanika.k@gmail.com','kanjun marg','2020-03-03',3000,'social media executive'),

(nextval('emp'),'sanjay tiwari',33,'8674342314','sanjay.t@gmail.com','kharghar','2018-04-01',3002,'Team Leader');

insert into employee(empid,empname,emp\_age,emp\_mobno,emp\_email,emp\_address,date\_of\_joining,emp\_deptid,emp\_designation)

values

(nextval('emp'),'sarika chakrapani',27,'7743422112','sarika211@gmail.com','sanpada','2012-04-01',3007,'Operations Executive'),

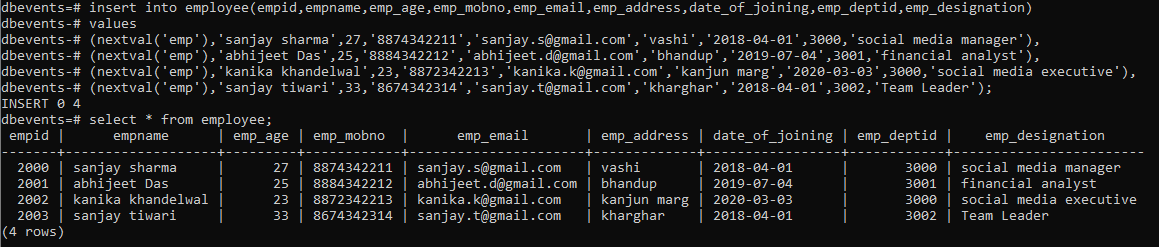
(nextval('emp'),'srinivas Jha',30,'8884343315','sri.j@gmail.com','vikhroli','2013-08-05',3005,'System analyst'),

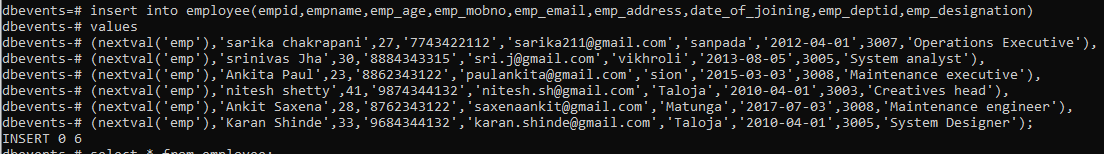
(nextval('emp'),'Ankita Paul',23,'8862343122','paulankita@gmail.com','sion','2015-03-03',3008,'Maintenance executive'),

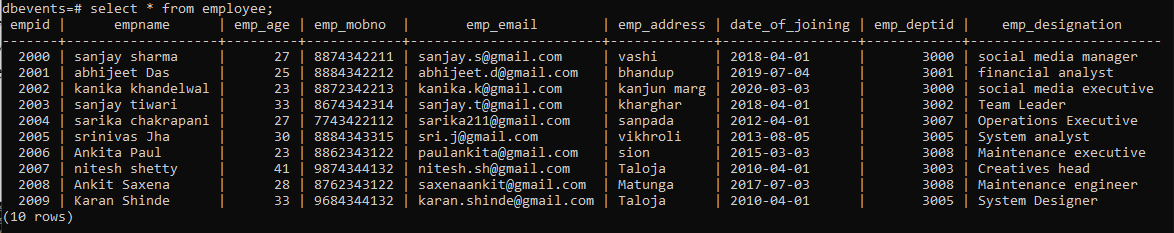
(nextval('emp'),'nitesh shetty',41,'9874344132','nitesh.sh@gmail.com','Taloja','2010-04-01',3003,'Creatives head'),

(nextval('emp'),'Ankit Saxena',28,'8762343122','saxenaankit@gmail.com','Matunga','2017-07-03',3008,'Maintenance engineer'),

(nextval('emp'),'Karan Shinde',33,'9684344132','karan.shinde@gmail.com','Taloja','2010-04-01',3005,'System Designer');







**3. Department:**

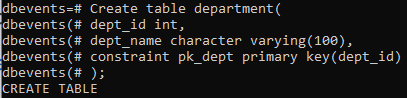
Create table department(

dept\_id int,

dept\_name character varying(100),

constraint pk\_dept primary key(dept\_id)

);

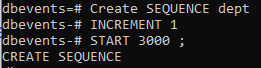


**Creating sequence for generating id:**

Create SEQUENCE dept

INCREMENT 1

START 3000 ;



**Inserting 5 records in department table**

insert into department(dept\_id,dept\_name)

values

(nextval('dept'),'Marketing'),

(nextval('dept'),'Finance'),

(nextval('dept'),'Production'),

(nextval('dept'),'Creative'),

(nextval('dept'),'HR');

insert into department(dept\_id,dept\_name)

values

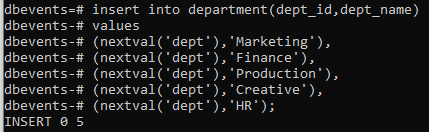
(nextval('dept'),'IT'),

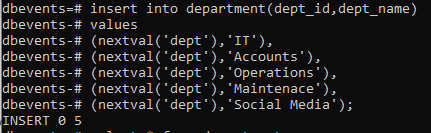
(nextval('dept'),'Accounts'),

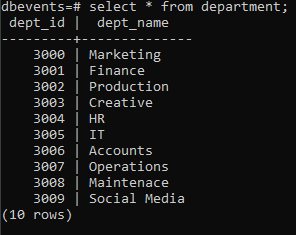
(nextval('dept'),'Operations'),

(nextval('dept'),'Maintenace'),

(nextval('dept'),'Social Media');







**4. Venue details:**

**Enum:**

Create type venue\_avl as enum ( 'avialable','not available');

Create table venue\_details(

venue\_id int,

venue\_name character varying(100),

no\_of\_seats int,

no\_of\_halls int,

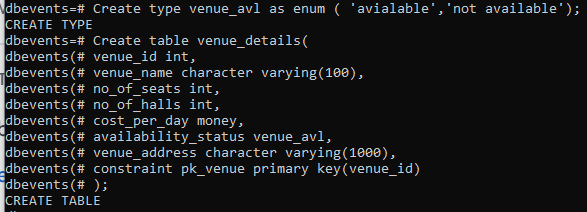
cost\_per\_day money,

availability\_status venue\_avl,

venue\_address character varying(1000),

constraint pk\_venue primary key(venue\_id)

);



Adding new value in enum.

ALTER TYPE venue\_avl ADD VALUE 'booked';

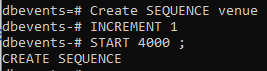


**Creating sequence for generating id:**

Create SEQUENCE venue

INCREMENT 1

START 4000 ;



**Inserting record in venue\_details:**

INSERT into venue\_details(venue\_id,venue\_name,no\_of\_seats,no\_of\_halls,cost\_per\_day,availability\_status,venue\_address)

values

(nextval('venue'),'celebrations hotel',4000,3,7000.00,'avialable','celebrations hotel vashi sec-29'),

(nextval('venue'),'Four Points by Sheraton',5000,5,12000.00,'avialable','Four Points by Sheraton vashi sec-27');

INSERT into venue\_details(venue\_id,venue\_name,no\_of\_seats,no\_of\_halls,cost\_per\_day,availability\_status,venue\_address)

values

(nextval('venue'),'pranam banquet',200,1,15000.00,'not available','airoli'),

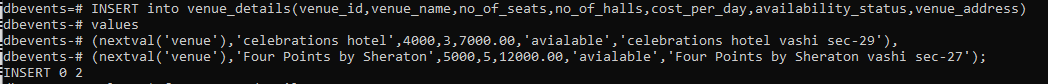
(nextval('venue'),'Orchids hotels',2000,2,25000.00,'avialable','nerul'),

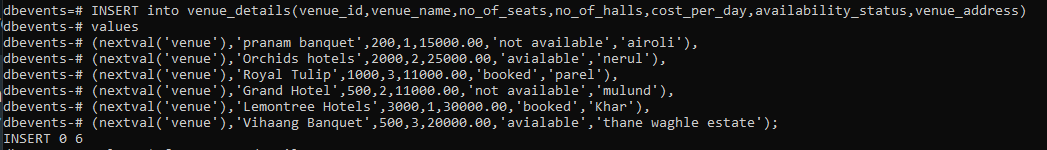
(nextval('venue'),'Royal Tulip',1000,3,11000.00,'booked','parel'),

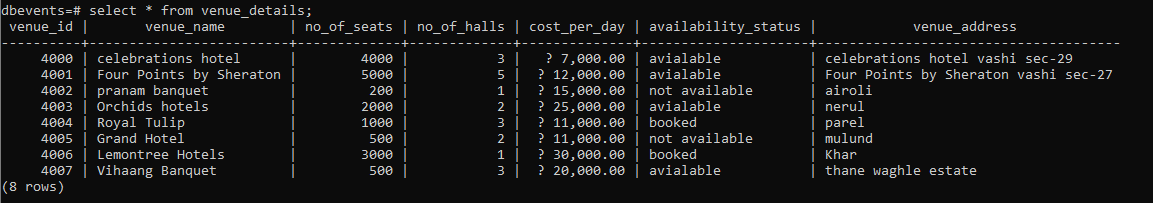
(nextval('venue'),'Grand Hotel',500,2,11000.00,'not available','mulund'),

(nextval('venue'),'Lemontree Hotels',3000,1,30000.00,'booked','Khar'),

(nextval('venue'),'Vihaang Banquet',500,3,20000.00,'avialable','thane waghle estate');







**5. Reservation\_details:**

Create type evt as enum ('offline','online','hybrid');

create table reservation\_detail(

r\_id int,

cust\_id int,

event\_name character varying(50) ,

date\_of\_event date,

venue\_id int,

event\_desc character varying(2000),

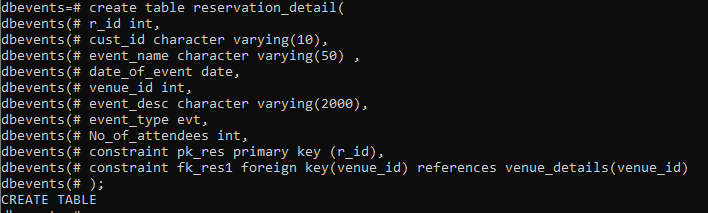
event\_type evt,

No\_of\_attendees int,

constraint pk\_res primary key (r\_id),

constraint fk\_res1 foreign key(venue\_id) references venue\_details(venue\_id)

);

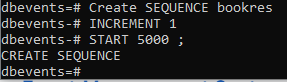


**Creating sequence for generating id:**

Create SEQUENCE bookres

INCREMENT 1

START 5000 ;



**Inserting records in reservation\_detail:**

Insert into reservation\_detail(r\_id,cust\_id,event\_name,date\_of\_event,venue\_id,event\_desc,no\_of\_attendees,event\_type)

VALUES

(nextval('bookres'),1000,'Birthday party','2021-07-27',4001,'Birthday party of Mehul Sharma',500,'offline'),

(nextval('bookres'),1002,'Skyesports awards','2021-06-12',4001,'SkyEsports award event',1000,'offline');

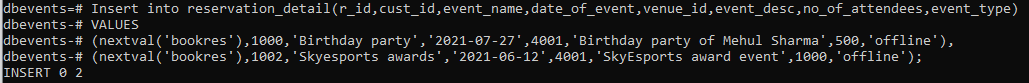
Insert into reservation\_detail(r\_id,cust\_id,event\_name,date\_of\_event,venue\_id,event\_desc,no\_of\_attendees,event\_type)

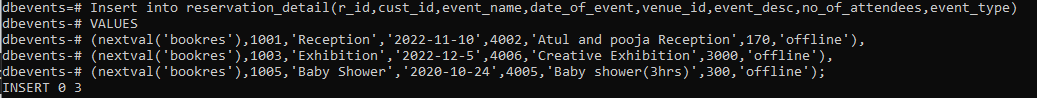
VALUES

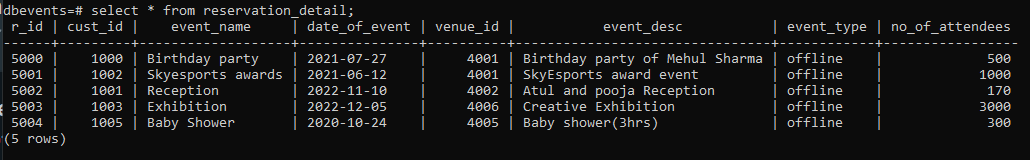
(nextval('bookres'),1001,'Reception','2022-11-10',4002,'Atul and pooja Reception',170,'offline'),

(nextval('bookres'),1003,'Exhibition','2022-12-5',4006,'Creative Exhibition',3000,'offline'),

(nextval('bookres'),1005,'Baby Shower','2020-10-24',4005,'Baby shower(3hrs)',300,'offline');







**6. Payment:**

Create table payment(

pid int primary key,

r\_id int,

payee\_name varchar(100),

payee\_accno varchar(100),

Amount money,

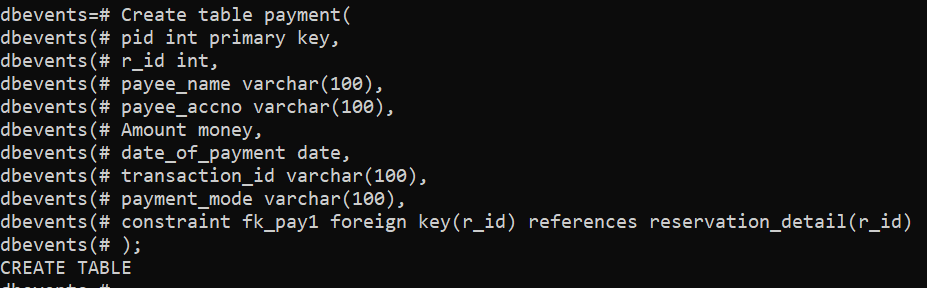
date\_of\_payment date,

transaction\_id varchar(100),

payment\_mode varchar(100),

constraint fk\_pay1 foreign key(r\_id) references reservation\_detail(r\_id)

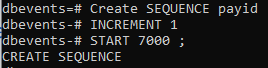
);



Create SEQUENCE payid

INCREMENT 1

START 7000 ;



**Inserting record into payment table:**

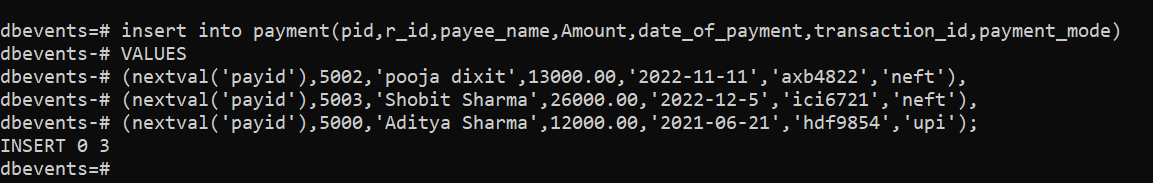
insert into payment(pid,r\_id,payee\_name,Amount,date\_of\_payment,transaction\_id,payment\_mode)

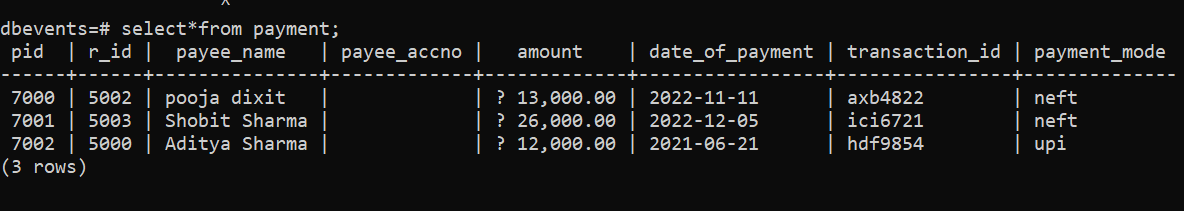
VALUES

(nextval('payid'),5002,'pooja dixit',13000.00,'2022-11-11','axb4822','neft'),

(nextval('payid'),5003,'Shobit Sharma',26000.00,'2022-12-5','ici6721','neft'),

(nextval('payid'),5000,'Aditya Sharma',12000.00,'2021-06-21','hdf9854','upi');





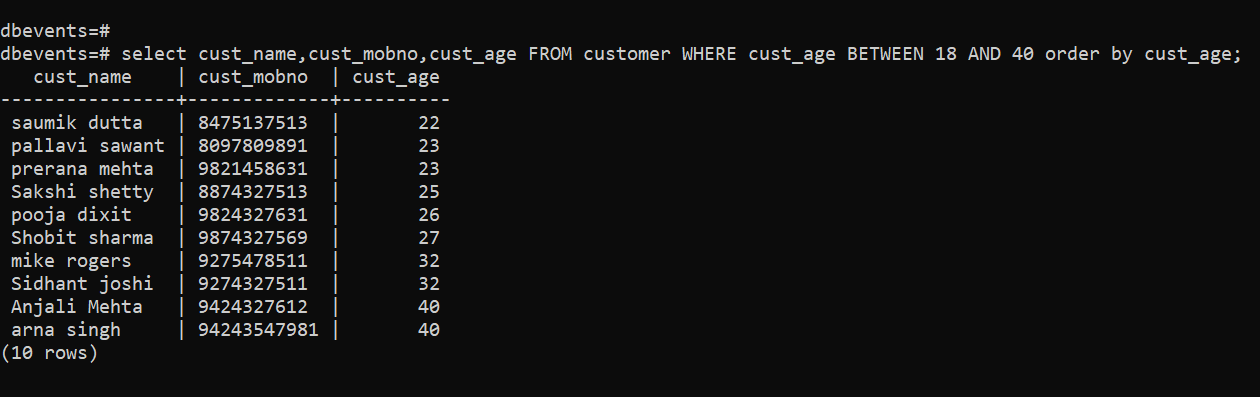
**QUERIES:**

**1) Fetch customer name and contactno of age between 18-40 in order**

**Query:-**

select cust\_name,cust\_mobno,cust\_age FROM customer WHERE cust\_age BETWEEN 18 AND 40 order by cust\_age;

**Result:-**



**2) Fetch records of customer names who have made reservations along with date of event and event name**

**Query:-**

select r.r\_id,c.cust\_id,c.cust\_name,r.event\_name,r.date\_of\_event

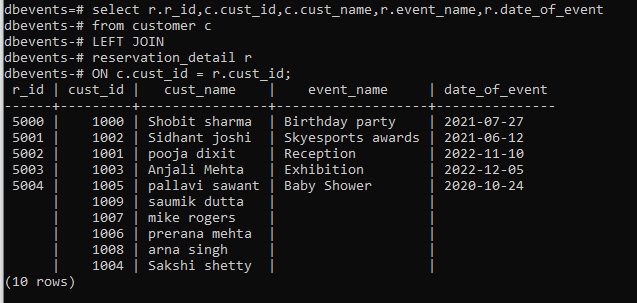
from customer c

LEFT JOIN

reservation\_detail r

ON c.cust\_id = r.cust\_id;

**Result:-**



**3) Fetch record of reservations with venue names and details**

**Reservation id, Event name,date, venue idvenue name ,venue address**

**Query:-**

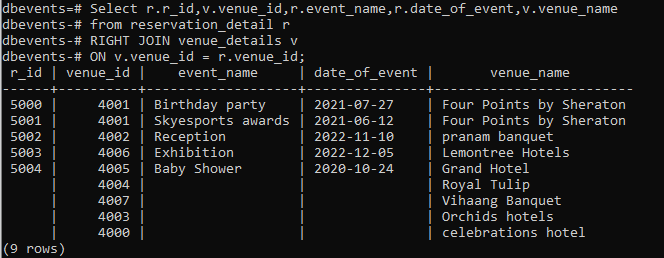
Select r.r\_id,v.venue\_id,r.event\_name,r.date\_of\_event,v.venue\_name

from reservation\_detail r

RIGHT JOIN venue\_details v

ON v.venue\_id = r.venue\_id;

**Result:-**

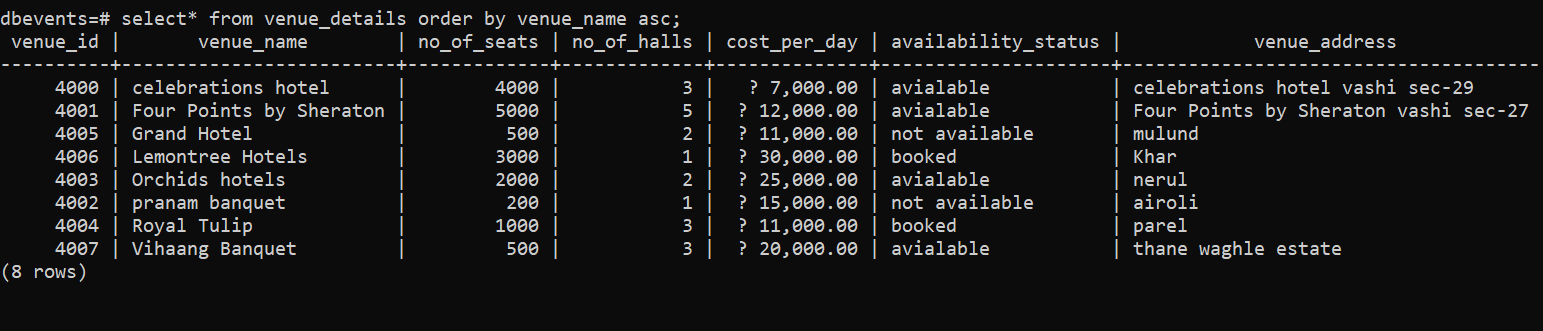


**4) Sort the venue details in ascending order:**

**Query:-**

select\* from venue\_details order by venue\_name asc;

**Result:-**

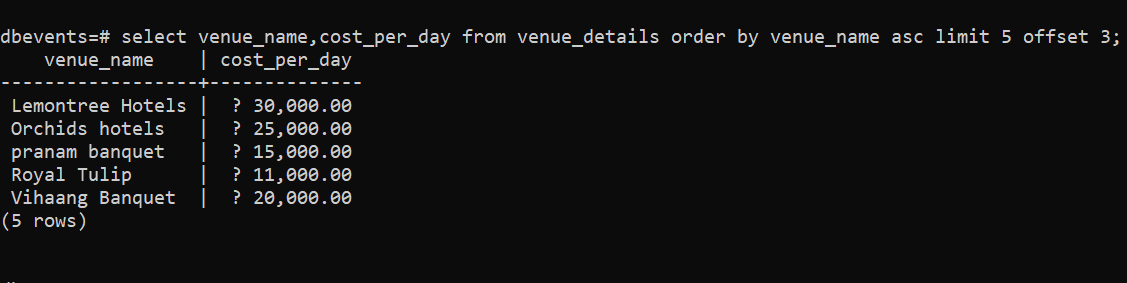


**5) Get the the five venue name,costperday from the sorted venuedetails after skipping first 3 records**

**Query:-**

select venue\_name,cost\_per\_day from venue\_details order by venue\_name asc limit 5 offset 3;

**Result:-**

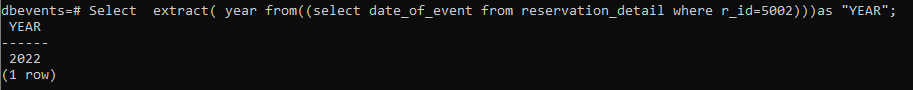


**6) Fetch the date of event year(only year) of reservation\_detail of all reservation when reservation id is given as input.**

**Query:-**

select extract( year from((select date\_of\_event from reservation\_detail where r\_id=5002)))as "YEAR";

**Result:-**

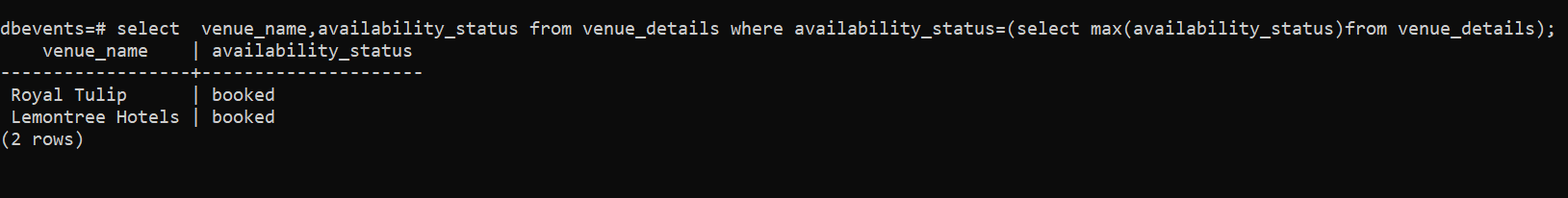


7) **Fetch records of availability status of enum greater than available**

**Query:-**

select venue\_name,availability\_status from venue\_details where availability\_status=(select max(availability\_status)from venue\_details);

**Result:-**

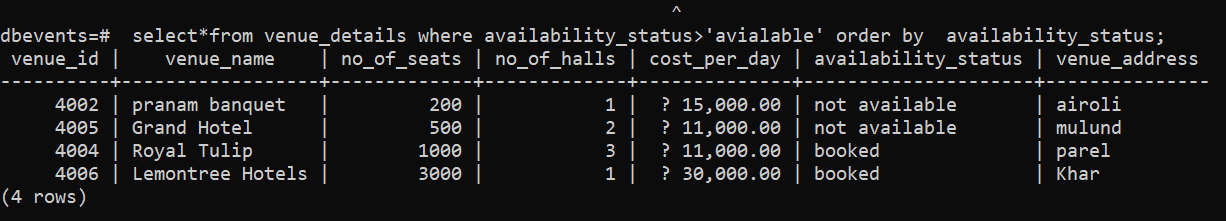
****

**8)Fetch records of availability status of enum greater than available**

**Query:-**

select\*from venue\_details where availability\_status>'avialable' order by availability\_status;

**Result:-**



**9) Fetch record of customers with payment and reservation details**

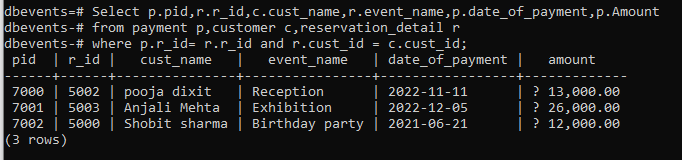
**Query:-**

Select p.pid,r.r\_id,c.cust\_name,r.event\_name,p.date\_of\_payment,p.Amount

from payment p,customer c,reservation\_detail r

where p.r\_id= r.r\_id and r.cust\_id = c.cust\_id;

**Result:-**



**10) Fetch the total amount paid by each customer**

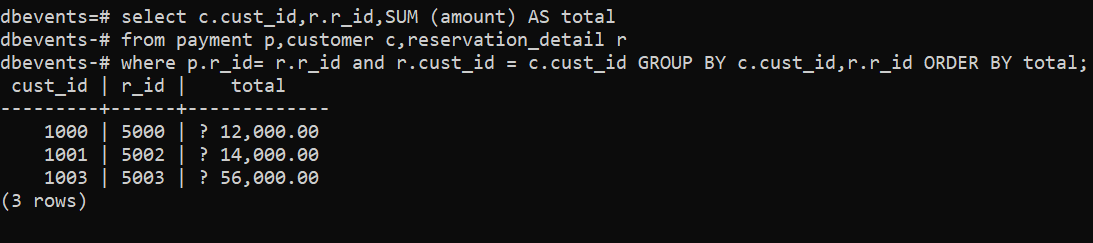
**Query:-**

select c.cust\_id,r.r\_id,SUM (amount) AS total

from payment p,customer c,reservation\_detail r

where p.r\_id= r.r\_id and r.cust\_id = c.cust\_id GROUP BY c.cust\_id,r.r\_id ORDER BY total;

**Result:-**



**11) Fetch Records of employees who are working under various departments.**

**Query:-**

Select e.empid,d.dept\_id,d.dept\_name,e.empname,e.emp\_designation

from employee e

JOIN

department d ON e.emp\_deptid=d.dept\_id;

**Result:-**

