# CSE5330: DATABASE SYSTEMS 1

# PROJECT 1

Name: Sai Kavya ,Dukkipati

UTA ID:1000980778

Task-1



LIST OF ENTITIES

* Postcard(POstcard)
* Condition(Condition\_PC)
* Time era(timeera)
* Color(Color)
* Collections
* Thematic category(thematiccategory)
* Thematic category and postcards(PCthem)
* Transactions (transactions)

B.

ATTRIBUTES AND THEIR DATATYPES

* Postcard(POstcard)-

pcID(INT):Postcard ID,

Title(varchar):Title of post card,

ConID(INT):Condition ID,

description(varchar):description of post card

* Condition(Condition\_PC)-

conID(int):Condition ID,

con(varchar):Condition

* Time era(timeera)-

eraID(int): Time era ID,

era(varchar):Time era

* Collections-

cpcid(int):Post card ID,

ccolid(int):Color ID,

ceraid(int):time era ID

* Thematic category(thematiccategory)-

themID(int):thematic category ID

them(varchar):thematic category

* Thematic category and postcards(PCthem)-

Pcid(int): post card ID

Themid(int):thematic category ID

* Transactions (transactions)-

tpcid(int) Post card ID

purchasedate(date): date of purchase

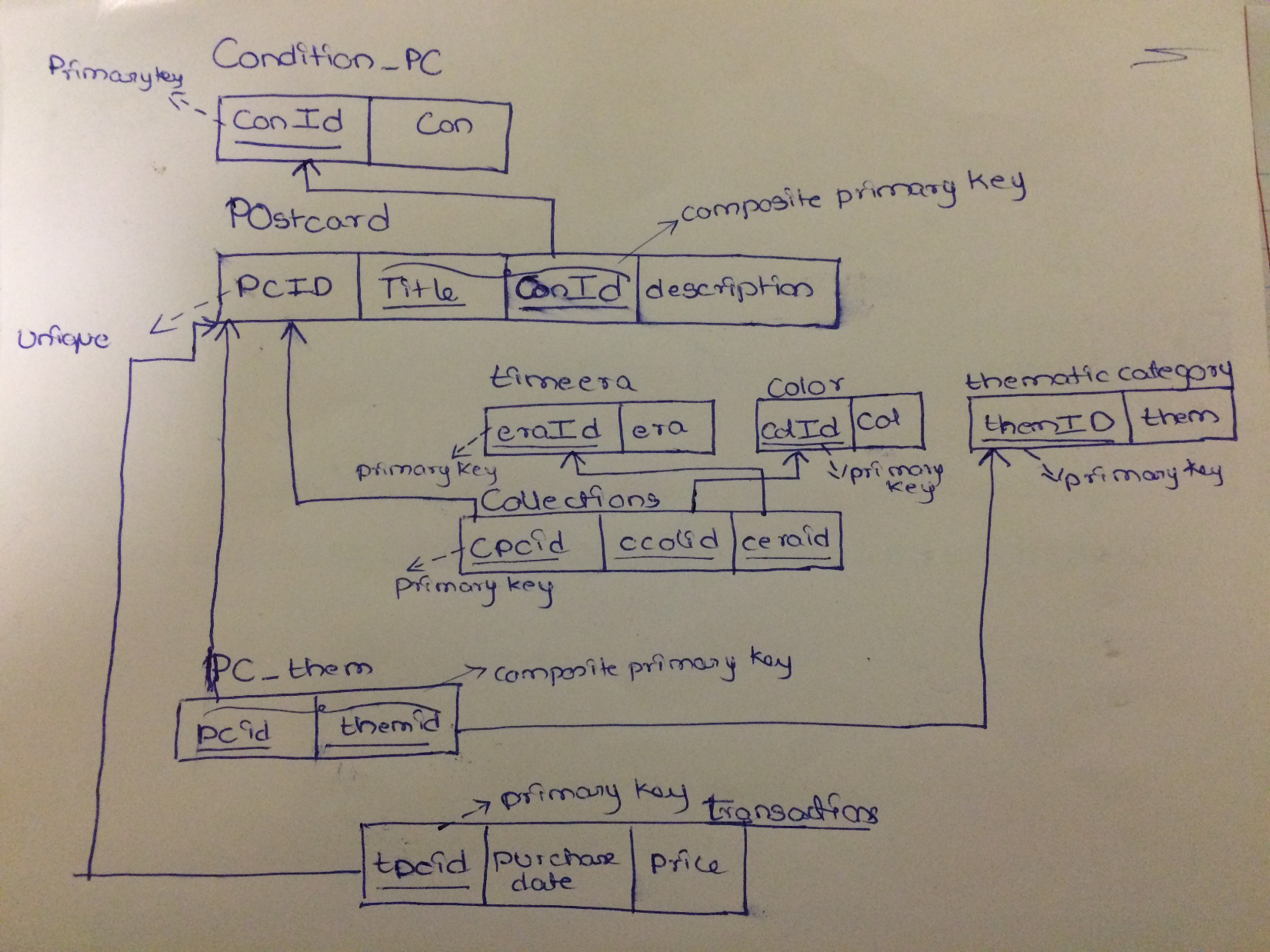
price(int):price of postcard

* Color(Color)-

colID(int):color ID

col(varchar): color

Task-2



A.

* As mentioned in the project description each post card can have multiple thematic categories so an entity “PC\_them” has been created where each post card can be mapped to different themes by making post card id and thematic category id as composite primary key. Cardinality of relation between post card and thematic category is one-to-many relation.
* There should not be multiple number of post cards having same condition so an entity has been created for post cards where each post card will have a unique ID and having title, condition as composite primary key so that no 2 post cards with same title will have same condition. Cardinality of relation between post card and condition is one-one relation.
* Transactions table has been created to track the date and price of purchase of post card.

1. The List of constraints posted on my schema are:

* Primary key constraint (ConID, eraID, colID, themID, tpcid, cpcid ) - maintains unique values, will not allow null values.
* Unique key constraint (pcID)-maintains unique values.
* Composite primary key ( (Title,conID),(pcID,them id)) – maintains unique combinations of attributes considered as composite primary key.
* Foreign key : Used maintain relations among tables .It is referred to parent table column which is primary key or unique .

Task-3

A.

* create table Condition\_PC(conID int,con varchar(25),primary key(conID));
* create table POstcard(pcID int NOT NULL UNIQUE,Title varchar(25),ConID int,description varchar(50),

foreign key(ConID) references Condition\_PC(conID),

Primary key(Title,ConID));

* create table timeera(eraID int,era varchar(25),primary key(eraID));
* create table color(colID int,col varchar(25),primary key(colID));
* create table thematiccategory(themID int,them varchar(25),primary key(themID));
* create table pcthem(themid int,pcid int,

foreign key(themid) references thematiccategory(themID),

foreign key(pcid) references POstcard(pcID),

primary key(themid,pcid));

* create table collections(cpcid int,ccolid int,ceraid int,

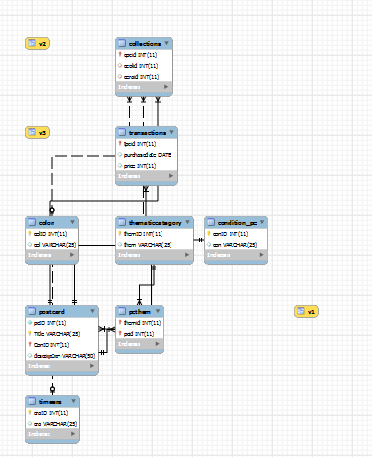
foreign key(cpcid) references POstcard(pcID),

foreign key(ccolid) references color(colID),

foreign key(ceraid) references timeera(eraID),primary key(cpcid));

* create table transactions(tpcid int,purchasedate date,price int,

foreign key(tpcid) references POstcard(pcID),primary key(tpcid));



B.

SQL code for inserting values into columns:

INSERT INTO

TABLE-NAME (ATTRIBUTE 1, ATTRIBUTE 2, ATTRIBUTE 3 …..)

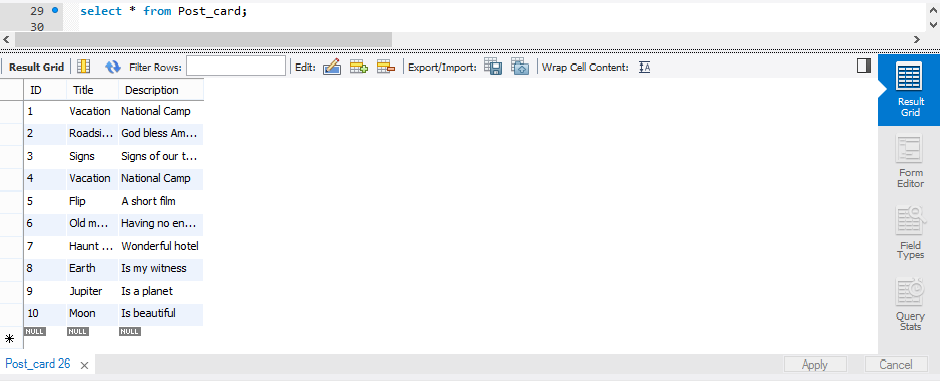
VALUES (VALUE 1, VALUE 2, VALUE 3 …….);

1. INITIAL STATE OF DATABASE :

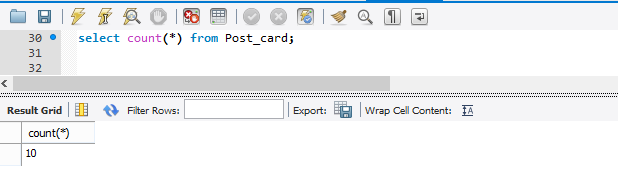
In the initial state the database will have Static entities like Time era, Color, Condition, Thematic category with data.

Task-4

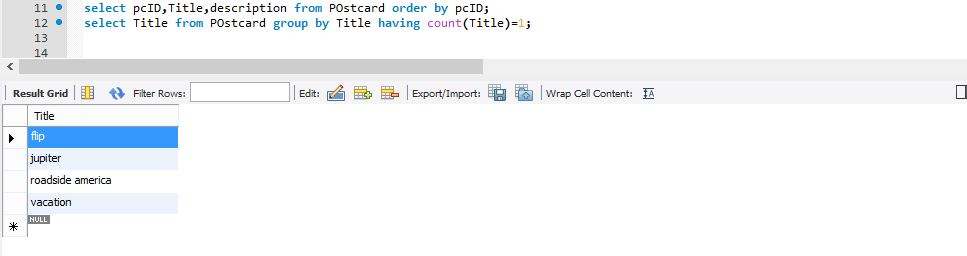
1. select \* from POstcard;



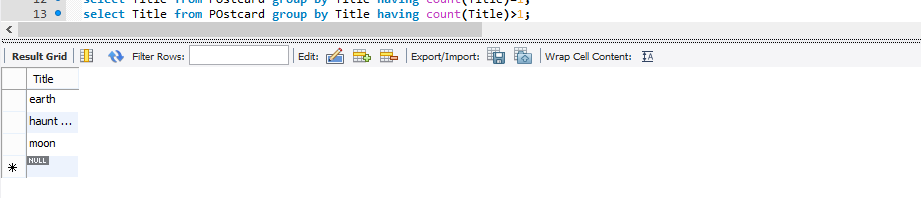
1. select count(\*) from POstcard;



1. select Title from POstcard group by Title having count(Title)=1;



1. select Title from POstcard group by Title having count(Title)>1;



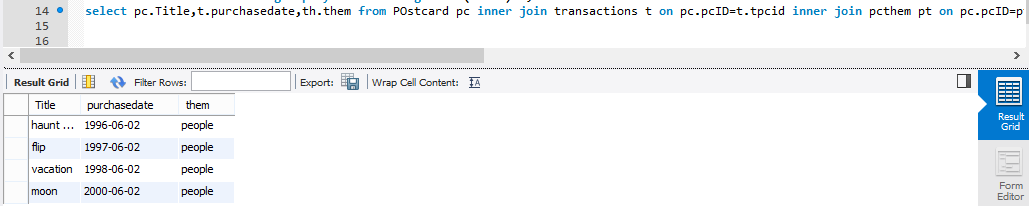
1. select pc.Title,t.purchasedate,th.them from POstcard pc

inner join transactions t on pc.pcID=t.tpcid

inner join pcthem pt on pc.pcID=pt.pcid

inner join thematiccategory th on pt.themid=th.themID

where t.purchasedate>1998/06/02 and th.them="people";



1. select pc.Title,e.era,th.them from POstcard pc

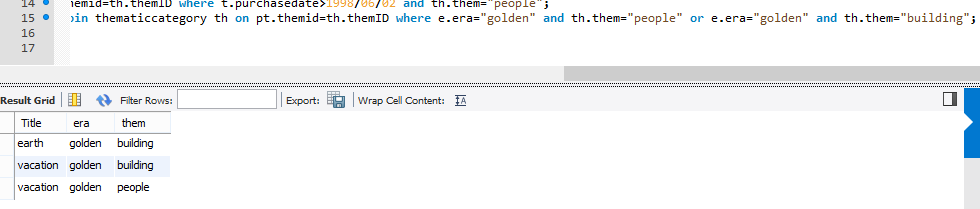
inner join collections c on pc.pcID=c.cpcid

inner join timeera e on c.ceraid=e.eraID

inner join pcthem pt on pc.pcID=pt.pcid

inner join thematiccategory th on pt.themid=th.themID

where e.era="golden" and th.them="people" or e.era="golden" and th.them="building";

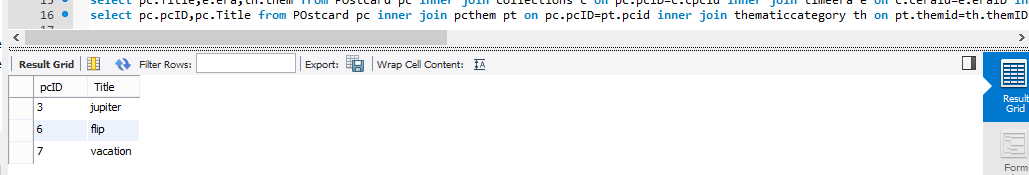


1. select pc.pcID,pc.Title from POstcard pc

inner join pcthem pt on pc.pcID=pt.pcid

inner join thematiccategory th on pt.themid=th.themID

group by(pc.pcID) having count(pc.pcID)>1;

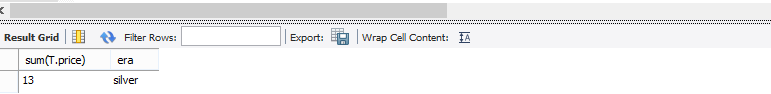


1. select sum(T.price),e.era from transactions T

inner join collections C on T.tpcid=C.cpcid

inner join timeera e on C.ceraid=e.eraID

where e.era="Silver";

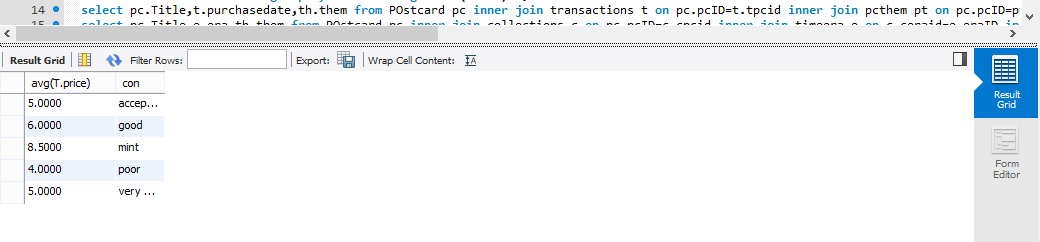


1. select avg(T.price),C.con from transactions T

inner join POstcard pc on T.tpcid=pc.pcID

inner join Condition\_PC C on pc.ConID=C.conID

group by C.con;

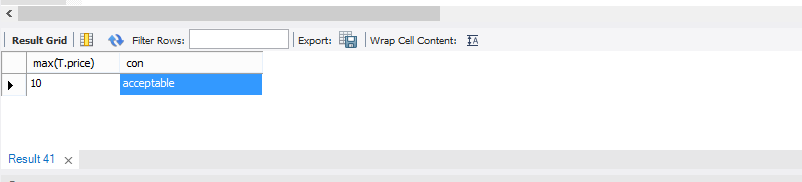


1. select max(T.price),C.con from transactions T

inner join POstcard pc on T.tpcid=pc.pcID

inner join Condition\_PC C on pc.ConID=C.conID

where C.con!="poor";



Task-5

1. create or replace view v1 as

select Title,Con,price from POstcard,Condition\_PC,transactions

where POstcard.pcID=transactions.tpcid

and POstcard.ConID=Condition\_PC.conID;

1. create or replace view v2 as

select Title,them,era from POstcard,thematiccategory,collections,timeera,pcthem where POstcard.pcid=collections.cpcid and POstcard.pcid=pcthem.pcid and pcthem.themid=thematiccategory.themID and collections.ceraid=timeera.eraID;

1. create or replace view v3 as

select avg(price),Title,them from transactions,POstcard,thematiccategory,pcthem

where transactions.tpcid=POstcard.pcID and POstcard.pcID=pcthem.pcid

and pcthem.themid=thematiccategory.themID group by them,title;