* Which of the following jobs would you be able to apply for if you gain skills in database systems? [2] **(a, c, d)**
  1. Database designer
  2. An app developer
  3. Database administrator
  4. SQL developer
* What is the name and version of the database management system we have been using in this course? [2] **(MySQL 8.0)**
* We have been studying relational databases throughout the semester. Why are they called relational databases? [3] **(discuss in class)**
* How do database management systems help us? Select all that apply**(b , d)**
  1. Help us keep our data in our file cabinet
  2. Provide data backup and data security
  3. Helps us write application code in PHP or Java
  4. Allows us to manage access to data by creating different levels of users with different permissions.
* What data type would you use for the primary key of a table that may contain 100 records at most? [1] **(unsigned tinyint)**
* What are the differences between DDL and DML? [3] **(creation vs manipulation - discuss)**
* State why database normalization is an important step in database design. [3] **(discuss in class)**
* We have the following table to keep pet information - [3]

**pets**(id, pet\_name, pet\_type, pet\_owner int) where the id has to be an integer type, pet\_name is text type, pet\_type can be dog, cat etc. and pet\_owner is an int column that actually points to the id of the owner whose information is in another table called owners.

This table can contain at most 10,000 pet records.

Show the table creation query for this table. Primary key, foreign key, nullability all should be taken into consideration. Must use the appropriate type for the id column to save space. (pk - unsigned smallint)

**Create table if not exists pets(id smallint unsigned auto\_increment primary key, pet\_name varchar (20) not null, pet\_type varchar(15) not null, pet\_owner int not null, foreign key(pet\_owner) references owners(id));**

* What will the following query show? Select \* from information\_schema.tables; [2] **(information about all the tables managed by this database server)**
* If I want to know everything about a table called *mytable,* what query should I run in MySQL? [2] **Select \* from information\_schema.tables where table\_name=’mytable’;**
* What is an ERD? [2] **(discuss in class)**
* What is that range of values for a column with unsigned mediumint? [2] **(0-16777215)**
* When should we use the enum data type in MySQL? [2] **(when a column can take value from a very limited number of unique values like days of the week, months, student status freshman, sophomore etc )**
* Declare a session variable called *actor* and set its value to “chaser”. Then run the query that will print “TOM” if the value of actor is chaser else print “JERRY”. Do not forget to use built-in functions for the purpose.[3]

**set @actor = 'chaser';**

**select if(@actor='chaser', 'TOM','JERRY');**

* Following is database schema -

**units**(id auto increment unsigned integer PK, Unit names (ST1 etc) varchar unique, unit\_height decimal, unit\_length decimal, unit\_width decimal, storage\_price decimal, rentable char (1))

**renters**(id auto increment unsigned integer PK, Renter name varchar(50), renter address varchar(100), renter phone varchar(15), renter email varchar (100), renter state id varchar (25), renter\_id\_type varchar(10), issuing\_state char(3))

**rentals**(id auto increment unsigned integer PK, rental start date date, rental end date date, storage unit (foreign key) referencing units, renter (FK) references renters)

What is the primary key in insurance types entity/table/relation? - use id for easy FK

**insurance\_types**(id auto increment unsigned integer pk, insurance type varchar(15), insurance price decimal)

**rental insurance**(id auto increment pk, insurance start date date, insurance end date date, rentals (FK) referencing rentals table, insurance (FK) references insurance\_types entity)

For marketing purposes, I need to know those renters (name and phone number) who are currently renting a unit but haven’t bought any insurance. Show the query that can give the data I need.

**Table creation queries -**

create table if not exists units

(id smallint unsigned auto\_increment primary key,

unit\_name varchar(4) unique, unit\_height decimal, unit\_length decimal,

unit\_width decimal, storage\_price decimal, rentable char (1) default 'Y');

create table if not exists renters

(id mediumint unsigned auto\_increment primary key,

renter\_name varchar(50) not null, renter\_address varchar(100), renter\_phone varchar(15) not null,

renter\_email varchar (100), renter\_state\_id varchar (25), renter\_id\_type varchar(10),

issuing\_state char(3));

create table if not exists rentals

(id integer unsigned auto\_increment primary key, rental\_start\_date date,

rental\_end\_date date, storage\_unit smallint unsigned, renter mediumint unsigned,

foreign key(storage\_unit) references units(id),

foreign key(renter) references renters(id)

);

create table if not exists insurance\_types

(id tinyint unsigned auto\_increment primary key,

insurance\_type varchar(15) not null,

insurance\_price decimal);

create table if not exists rental\_insurance

(id integer unsigned auto\_increment primary key,

insurance\_start\_date date, insurance\_end\_date date,

rental\_id integer unsigned not null,

insurance\_type tinyint unsigned,

foreign key(rental\_id) references rentals(id),

foreign key(insurance\_type) references insurance\_types(id));

–

Populate data

–

select renter\_name, renter\_phone from renters

where renters.id not in (

select r.id from renters r

join rentals rl

on rl.renter=r.id

join rental\_insurance ri

on

ri.rental\_id=rl.id); **Is there something wrong with this query? Does it show data by the renter, not by rental? If so, what is wrong with that? How can we make it better?**

* Following is a database schema for storage rental business -

units(id auto increment unsigned integer PK, Unit names (ST1 etc) varchar unique, unit\_height decimal, unit\_length decimal, unit\_width decimal, storage\_price decimal, rentable char (1))

renters(id auto increment unsigned integer PK, Renter name varchar(50), renter address varchar(100), renter phone varchar(15), renter email varchar (100), renter state id varchar (25), renter\_id\_type varchar(10), issuing\_state char(3))

rentals(id auto increment unsigned integer PK, rental start date date, rental end date date, storage unit (foreign key) referencing units, renter (FK) references renters)

insurance\_types(id auto increment unsigned integer pk, insurance type varchar(15), insurance price decimal)

rental\_insurance(id auto increment pk, insurance start date date, insurance end date date, rentals (FK) referencing rentals table, insurance (FK) references insurance\_types entity)

Write a procedure that will accept an insurance type ( the name of the insurance, not the id) as an input parameter and check if there is any rental associated with that insurance at all. The procedure will follow the steps given below to do this -

Step 1: check if a valid insurance type was given by checking if that insurance type exists in the insurance\_types table. Insurance types can be - FIRE, FLOOD, EARTHQUAKE. If the insurance does not exist, then it will throw an error saying "Invalid insurance type." If the insurance type exist, then it will go to step 2.

Step 2: Now it will hit rental\_insurance table to see if there is any record of that insurance type. If there is, then the procedure will return the word "YES", else it will return the word "NO."

You will get partial marks. So try to answer as best as you can.

* Following is a database schema for storage rental business -

**units**(id auto increment unsigned integer PK, Unit names (ST1 etc) varchar unique, unit\_height decimal, unit\_length decimal, unit\_width decimal, storage\_price decimal, rentable char (1))

**renters**(id auto increment unsigned integer PK, Renter name varchar(50), renter address varchar(100), renter phone varchar(15), renter email varchar (100), renter state id varchar (25), renter\_id\_type varchar(10), issuing\_state char(3))

**rentals**(id auto increment unsigned integer PK, rental start date date, rental end date date, storage unit (foreign key) referencing units, renter (FK) references renters)

**insurance\_types**(id auto increment unsigned integer pk, insurance type varchar(15), insurance price decimal)

**rental\_insurance**(id auto increment pk, insurance start date date, insurance end date date, rentals (FK) referencing rentals table, insurance (FK) references insurance\_types entity)

Write a function that will take a phone number as an input and then find out how many rentals are associated with that phone number. It will return the number of rentals.