-------RECORD QUESTIONS-----------

Normalization Example

DETAILED HEALTH HISTORY REPORT

PET ID	PET NAME	PET TYPE	PET AGE	OWNER	VISIT DATE	PROCEDURE
246	ROVER	DOG	12	SAM COOK	JAN 13/2013	01 - RABIES VACCINATION
					MAR 27/2013	10 - EXAMINE and TREAT WOUND
					APR 21/2013	05 - HEART WORM TEST
298	SPOT	DOG	2	TERRY KIM	JAN 21/2013	08 - TETANUS VACCINATION
					MAR 10/2013	05 - HEART WORM TEST
341	MORRIS	CAT	4	SAM COOK	JAN 23/2012	01 - RABIES VACCINATION
					JAN 13/2013	01 - RABIES VACCINATION
519	TWEEDY	BIRD	2	TERRY KIM	APR 21/2013	20 - ANNUAL CHECK UP
					APR 30/2013	12 - EYE WASH

BASIC MySQL commands

1.command to start MySql database server

2.create a database called "MCAdemo"

3.command to find out databases exists on the server

4.command that specifies which database should be made active

5.command to destroy database

6.command to verify support for the various MySQL table data storage

7.command to exit from the MySql prompt and return to the system prompt.

TABLE CREATION WITHOUT CONSTRAINTS

1.Create the following table called Pet

ColName	Data type	width
Name	String	20
Owner	String	20
Species	String	10
sex	String	1
Birth	Date	

Death	Date	

2.

(a) to add new records one at a time, use INSERT

NAME	OWNER	SPECIES	SEX	BIRTH	DEATH
Fluffy	Harold	cat	f	1993-02-04	
Claws	Gwen	cat	m	1994-03-17	
Buffy	Harold	dog	f	1989-05-13	
Fang	Benny	dog	m	1990-08-27	
Bowser	XDiane	dog	m	1979-08-31	1995-07-29
Chirpy	Gwen	bird	f	1998-09-11	
Whistler	Gwen	bird		1997-12-09	
Slim	Benny	snake	m	1996-04-29	

3. Retrieving Information from a Table

```
SELECT what_to_select
FROM which_table
WHERE conditions_to_satisfy;
```

3.1 : Selecting All Data

3.2 Slecting a particular row

3.2.1. select Bowser's record

3.2.2 select animals born during or after 1998

3.3. Combining conditions using AND,OR

3.3.1 select query to locate female dogs

3.3.2 select birds and snakes from the table

3.3.3 select male cats and female dogs from the table

3.4 selecting particular columns

3.4.1 select only name,birth from table pet

3.4.2 list all owners of pets from the table

3.5 To eliminate duplicate entires(or to list only unique entries) use DISTINCT

3.6 Sorting rows : ORDER BY

3.6.1 Ascending order of date of birth

3.6.2 Descending order of date of birth

3.6.3 Sort by type of animal in ascending order, then by birth date within animal type in descending order

4.Updating records

```
UPDATE table_name
SET
  column_name1 = expr1,
  column_name2 = expr2,
  ...
[WHERE
  condition];
```

Notice that the `WHERE` clause is so important that you should not forget. Sometimes, you may want to change just one row; However, you may forget the `WHERE` clause and accidentally update all rows of the table.

4.1: Update the sex for Whistler as male.

4.2: Update year of death of all pets to 2002-01-17 except Bowser.

4.3: Update year of death of all pets to 2004-04-14

4.4: Update owner of male dogs to Benny

4.5: Update species=cat and sex=m for Chirpy.

5.Deleting Records

```
DELETE FROM `table_name` [WHERE condition];
```

- DELETE FROM `table_name` tells MySQL server to remove rows from the table ..

- [WHERE condition] is optional and is used to put a filter that restricts the number of rows affected by the DELETE query.

5.1: Delete all records owned by harold.

5.2:Delete all female birds.

5.3:Delete all records whose year of birth is after 1994.

5.4:Delete all records from table pet.

6. Altering table structure

a) For adding new column

```
ALTER TABLE table_name
ADD column_name datatype;
```

b)For droping a column

```
ALTER TABLE table_name
DROP COLUMN column_name;
```

c)To modify a column

```
ALTER TABLE table_name
MODIFY COLUMN column_name datatype;
```

6.1: Add a column "weight " of datatype integer to the pet table.

6.2: Add a column "id " of datatype integer to the pet table(as first column)

6.3:Add a column "breed" of datatype string of size 10 to the pet table(after the column death)

6.4:Modify column "breed" of datatype string to 20 in pet table.

6.5:Modify(shift) "breed" the column immediately after column weight.

6.6:Rename column name "pet" to "pet_id".

6.7: Rename "weight" to "pet_weight" and shift immediately after "breed".

6.8: Drop column "breed"

6.9:Rename table "pet" to "pet_info"

```
ALTER TABLE table_name
RENAME TO new_table_name;
```

6.10:Empty table pet using truncate

The SQL TRUNCATE TABLE statement is used to remove all records from a table. It performs the same function as a DELETE statement without a WHERE clause.

```
TRUNCATE TABLE table_name;
```

6.11: Drop table structure along with the data of table pet_table.

7 Using arithmetic operators in SQL statements

Numerous conditional restrictions can be placed on the selected table contents in the WHERE clause of the SELECT statement. For example, the comparison operators shown in Table can be used to restrict output

SYMBOL	**MEANING**
=	Equal to
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
<> or !=	Not equal to
BETWEEN	Used to check if an attribute is within a range.
IN	Used to check if an attribute value matches any value within a list.
LIKE	Used to check if an attribute value matches a given string pattern.
IS NULL / IS NOT NULL	Used to check if an attribute is NULL / is not NULL.

CREATE TABLE TICKET

TICKET_NO	TICKET_PRICE	TICKET_TYPE	PARK_CODE
13001	19	Child	FR1001

13002	35	Adult	FR1001
13003	21	Senior	FR1001
88567	23	Child	UK3452
88568	42	Adult	UK3452
89720	11	Senior	UK3452

7.1 : compare the current ticket prices, with an increase in the price of each ticket by 10%.

7.2: display the theme park code, ticket price and ticket type of all tickets where the ticket price is greater than Rs.20.00.

7.3: display tickets that are less than Rs.30.00.

7.4:list all tickets whose prices are between Rs.30 and Rs.50

7.5: Write a query to display all tickets that are of type Senior or Child

LIKE

The LIKE operator is used to find patterns within string attributes. Standard SQL allows you to use the percent sign (%) and underscore (_) wildcard characters to make matches when the entire string is not known. % means any and all following characters are eligible while _ means any one character may be substituted for the underscore.

7.6: Find all pet rows whose first names begin with the letter f.

7.7: Find all Theme Parks that have a name ending in ‘er’.

7.8 Find all pet names whose second letter is “h”

8.**Using Logical Operators**

8.1 List all Ticket details price is greater than 20 and park_code is “uk3452”.(use ticket table)

8.2 List all Ticket details for adults and seniors

8.3List all ticket details except “88568”

9.**Functions in MySQL**

Aggregate Functions

9.1 Find count of Dogs from pet table**.**

9.2 Query which illustrates the use of MIN and MAX of ticket price.

9.3 Query which illustrates the use of SUM and AVG of ticket price**.**

String Functions

9.4 Concatenation

9.5Length

9.6 Substr

9.7 Lower

9.8 Upper

Date Functions

9.10 Month

9.11Dayofmonth

9.12 year

9.13 datediff

9.14 date_Add

9.15 Date_sub

9.16 Last_Day

Numeric Functions

9.17 abs()

9.18 round()

9.19 truncate()

9.20 mod()

10. Group by and Having

10.1 List the number of pets owned by each person(owner) from pet table.

10.2List pets belonging to a particular species.

10.3 List number of pets based on their sex.

10.4 List the number of pets owed by each owner from the table (list only those owners having 2 or more pets)

11.**Table creation with constraints**

11.1

Aud_id	Integer	Primary key
Auth_name	String	Not null
Auth_email	String	Unique
Country	String	Default value=”india”
Homecity	String	Default value=”Delhi”
No_of_publications	Integer	>=5

11.2 Create a table book with following fields

COLUMN NAME	TYPE	CONSTRAINT
Book_id	Integer	Primary key,auto_increment
Title	String	Not null
Auth_id	Integer	Foreign key

11.3 Foreign key constraint with on delete cascade

Building

Room

Building_no	Int	PK		room_no	Int	PK
Building_name	Varchar			room_name	Varchar	
Address	varchar			Building_no	Int	FK

11.4 Create above tables using Foreign key constraint with on delete set null.

12. **Alter table with constraints**

Primary Key Using ALTER TABLE Statement

ALTER TABLE table_name **ADD PRIMARY KEY**(column_list);

Drop primary key

ALTER TABLE table_name DROP PRIMARY KEY;

Add unique key

-
  ```
  ALTER TABLE Persons
  ADD UNIQUE (ID);
  ```

-
  ```
  ALTER TABLE Persons
  ADD CONSTRAINT UC_Person UNIQUE (ID,LastName);
  ```

Drop unique key

```
ALTER TABLE Persons
DROP INDEX UC_Person;
```

Add Foreign key

```
ALTER TABLE Orders
ADD CONSTRAINT FK_PersonOrder
FOREIGN KEY (PersonID) REFERENCES Persons(PersonID);
```

Drop

```
ALTER TABLE Orders
DROP FOREIGN KEY FK_PersonOrder;
```

Create table Persons

P_id	Integer
First_name	String
Last_name	String
Address	String

12.1 Add primary key constraint on pid;

12.2 Drop primary key constraint

12.3 Add primary key constraint on multiple coloumn(pid,firstname)

12.4 Add unique constraint on pid column

12.5 Add unique constraint on firstname and lastname column.

12.6 Drop unique constraint

Create table Order

O_id	Integer
Order_no	Integer
Pid	Integer

12.7 To add foreign key constraint on the pid.

12.8Drop a foreign key constraint.

13.Joins

- create *suppliers* with two fields (supplier_id and supplier_name). It contains the following data:

supplier_id	supplier_name
10000	IBM
10001	Hewlett Packard
10002	Microsoft

10003	NVIDIA

- Create table called *orders* with three fields (order_id, supplier_id, and order_date). It contains the following data:

order_id	supplier_id	order_date
500125	10000	2013/05/12
500126	10001	2013/05/13
500127	10004	2013/05/14

Inner join

```
SELECT
t1.id, t2.id
FROM
t1
INNER JOIN
t2 ON t1.pattern = t2.pattern;
```

LEFT OUTER JOIN

```
SELECT columns
FROM table1
LEFT [OUTER] JOIN table2
ON table1.column = table2.column;
```

RIGHT OUTER JOIN

```
SELECT columns
FROM table1
RIGHT [OUTER] JOIN table2
ON table1.column = table2.column;
```

CROSS JOIN

```
SELECT
t1.id, t2.id
FROM
t1
CROSS JOIN t2;
```

13.1 List details of suppliers(along with order date), who supplies products corresponding to the orders in order table.

13.2 List details of all suppliers in the "supplier" table along with order details(if any) handled.

13.3 List details of all Orders in the "order" table along with suppliers details(if any).

13.4 List all possible combinations of order, supplier details

14. SET OPERATIONS

Create tables Course1

ID	NAME
1	Jack
2	Harry
3	Jackson

Create table Course2

ID	NAME
3	Jackson
4	Stephan
5	David

14.1 List details of all students registered for atleast one course(course1 or course2 or both).(without duplicates)
14.2 List details of all students registered for atleast one course(course1 or course2 or both).(with duplicates)

15.SUBQUERIES

Sample table: Salesman

salesman_id	name	city	commission
5001	James Hoog	New York	0.15
5002	Nail Knite	Paris	0.13
5005	Pit Alex	London	0.11
5006	Mc Lyon	Paris	0.14
5003	Lauson Hen	San Jose	0.12
5007	Paul Adam	Rome	0.13

Sample table: Orders

ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.5	2012-10-05	3005	5002
70009	270.65	2012-09-10	3001	5005
70002	65.26	2012-10-05	3002	5001
70004	110.5	2012-08-17	3009	5003
70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.6	2012-04-25	3002	5001

15.1. Write a query to display all the orders from the orders table issued by the salesman 'Paul Adam'.

15.2 Write a query to find all the orders issued against the salesman who may works for customer whose id is 3007.

15.3 Write a query to display all the orders which values are greater than the average order value for 10th October 2012

15.4 Write a query to display the commission of all the salesmen servicing in Paris.

15.5 Write a query to display all the customers whose id is 2001 bellow the salesman ID of Mc Lyon

16.VIEWS

customer_id	cust_name	city	grade	salesman_id
3002	Nick Rimando	New York	100	5001
3007	Brad Davis	New York	200	5001
3005	Graham Zusi	California	200	5002
3008	Julian Green	London	300	5002
3004	Fabian Johnson	Paris	300	5006
3009	Geoff Cameron	Berlin	100	5003
3003	Jozy Altidor	Moscow	200	5007

3001 | Brad Guzan | London | 100 | 5005

16.1 Write a query to create a view "highgrade" that shows all of the customers who have grade 300.

16.2 Alter the view "highgrade" that shows all of the customers who have the grade >=200.

16.3 Update city of customer with salesman_id=5006 to "London" in the view " highgrade "

16.4 Delete customer details from view " highgrade" 17.5.Drop view " highgrade "

17.Backing Up and Restoring MySQL database

The mysqldump command is used to create textfile “dump” of a database that can be managed by MySQL. These database dumps are simply text type files containing all the SQL commands needed to recreate the database from scratch. The process truly is quick and easy.

mysqldump -u [username] -p [databasename]>[backupfile.sql]

Eg: mysqldump -u root -p demo>demo.sql;

Restoring backup

Since the dump files are just data preceded by SQL commands, you can restore the database backup by telling MySQL to run the commands in it and put the data back into the proper database.

Eg:**mysql -u root -p demo1<demo.sql**

18.Stored Procedure

18.1 Write a procedure to input a single digit number and convert it into words.

18.2 Write a procedure that will accept a ticket number from the user.Check if the price is greater than the minimum (Rs.20) .The n deduct Rs.5 from the price.

19.Stored Functions

19.1 Write a stored function ,take input parameter as mark and return distinction,first class ect.

19.2 Call a stored function within a stored procedure.(Write a function to Check odd or even,Call this function in a procedure)

19.3 Use employee database.Write a function that returns the number of years the employee has been in the company.(Create an emp table with id,name,dateofjoin,salary)

20.Triggers

create table emp_log(id int ,last_modified_time datetime)

20.1

(a)Before Insert triggers insert values into employee table salary amount is to be set as (salary+100) using triggers.

20.2 Before update triggers

create a trigger on employee on updating

if salary <500;set salary=salary+100

elseif salary<1000;set salary=salary+500

else salary=salary+1000

20.3 After insert triggers

Create trigger bi_emps after insert on emps for each row insert required fields into emp_log table.

20.4 After Update triggers

Create trigger aftr_upd_emps after update on emps for each row

update required fields of emp_log table.

21.Cursors

21.1 Create a cursor for reading student mark calculation

(Create a table student with id,name,mark1,mark2,mark2.,total,average and grade.

insert only id,name marks.calculate and update total,avg and grade)

21.2Create a cursor for reading total mark from student table and insert id,names into pass table and fail table respectively.