Alter table with constraints

Delete table

Create table with constraints

**MySQL CONSTRAINTs are :**<https://www.w3resource.com/mysql/creating-table-advance/constraint.php>

* NOT NULL
* UNIQUE
* PRIMARY KEY
* FOREIGN KEY
* CHECK
* DEFAULT

Create table customers ( cust\_id int primary key,

                                        cust\_name varchar(20) not null,

                                         cust\_email varchar(20) unique);

Insert into customers values(1, ‘john’,’[john@gmail.com](mailto:john@gmail.com)),

                                            ( 2, ‘jane’, ‘[jane@gmail.com](mailto:jane@gmail.com)’),

                                            ( 3, ‘jake’,’[jake@gmail.com](mailto:jake@gmail.com)’);

Insert into customers (cust\_id, cust\_name) values (4, ‘joseph’);

Update customers set cust\_email=’[joseph@gmail.com](mailto:joseph@gmail.com)’ where id=4;

-- deletes all the rows

Delete from customers;

-- delete a particular row when the condition is specified

Delete from customer where id = 2;

-- all records are committed by default.

- my sql transaction are commited

Select @@autocommit;    cannot be rolled back

Commit value can be changed

Set autocommit =0;

Select \* from customers;

Delete from customers;

Rollback;

Select \* from customers;

Any dml commands can be got back using rollback.

Delete

* Delete all records from table when condition is specified
* Specify condition to delete particular records
* You can rollback the transaction
* Fire triggers before you delete (what is trigger --- are any event happens it will trigger the particular command)

Truncate

* Delete all the records from the table
* -there is no option to provide where condition it will delete all the records
* In truncate you cannot rollback it is for all ddl statement
* Delete executes slower compared to truncate delete uses cache which enables us to rollback. Since truncate cannot be rolled back it does not cache so it is faster

Dml

Delete from customers where id = 2;

ddl

Truncate

Drop - will delete all record and also drop the table

drop table table\_name;

Drop

Drop delete all records along with the schema or structure of the table.

So you cannot use the table. You need to create again.

Drop table customers;

Select \* from customers;

---table does not exist----

The above are important interview questions

DATABASE OBJECTS

Objects which are physically existing in the db.

Tables, stored procedures, functions, triggers, views, etc.

Table - a physical structure to store data. Contains rows and columns. Rows are  interrelated data/information, columns are list of same information

St pro - set of commands

Usually we right inline commands. In stored procedures, we give a set of commands in the required order of execution to get the desired result.

Click stored pro -- create store procedure

Create procedure ‘getcustomers‘()

Begin

     Select \* from customers;

End

Call getcustomers()

Create procedure ‘getcustomers‘( id int)

Begin

Select \* from customers where cust\_id = id;

End

Call getcustomers(2);

Create procedure ‘getcustomers‘( id int)

Begin

Select \* from customers;

Select \* from customers where cust\_id = id;

End

create procedure getAmount(unitPrice p, quantity q, amount amt)

begin

.....

amt = p \* q;

end

create procedure invoice()

begin

...

--> price= 1.25 , qty = 100, amount =0

call getAmount( price, qty, amount)

select amount

end

You can use for insert, update

Compared to inline commands stored procedure is better.

First time it complies second time it only executes

So faster and secure than inline commands

Accessibility can be controlled for the operations.

Overview

Performance is good

Once created sp are compiled and stored in database.

Functions also execute a set commands/queries

Functions  -   Date Function

CREATE DEFINER=`root`@`localhost` FUNCTION `getdatebasedonindex`(

dtDate DATETIME,

   intMonthIndex INT

) RETURNS date

BEGIN

RETURN

   DATE\_SUB(

       LAST\_DAY(

           DATE\_ADD(dtDate, INTERVAL intMonthIndex MONTH)

       ),

       INTERVAL DAY(

           LAST\_DAY(

               DATE\_ADD(dtDate, INTERVAL intMonthIndex MONTH)

           )

       )-1 DAY

   );

END

Select getdatebasedonindex(‘2017-04-04’, 2)

CREATE DEFINER=`root`@`localhost` FUNCTION `upperString`(str varchar(15)) RETURNS varchar(30) CHARSET utf8

BEGIN

RETURN concat('to upper = ', upper(str));

END

select upperString(cust\_name) from customers;

Reusable , returns only one value   Routines -- any set of statements to be executed it can be functions or procedures

<https://dev.mysql.com/doc/refman/5.7/en/stored-routines-syntax.html>

Triggers - based on event triggers can be fired

* A trigger is a named database object that is associated with a table, and that activates when a particular event occurs for the table. Some uses for triggers are to perform checks of values to be inserted into a table or to perform calculations on values involved in an update.
* <https://dev.mysql.com/doc/refman/5.7/en/triggers.html>

Triggers are called before an event

* Events which fires before/after crud operations

Countries-before insert  audit table hr audit to keep track of all the transactions

amazon

sony erricson - db2

lenova -- db

hp --- database

VIEWS

* Virtual table
* Any update or delete command on the view affects the main table
* Rules to follow are the same constraints to be set in view
* Referential integrity constraint in the main table then view must contain that

Column and proper value must be provided to that column.

CREATE

   ALGORITHM = UNDEFINED

   DEFINER = `root`@`localhost`

   SQL SECURITY DEFINER

VIEW `cust` AS

   SELECT

       `customers`.`Cust\_id` AS `Cust\_id`,

       `customers`.`Cust\_name` AS `Cust\_name`,

       `customers`.`Cust\_email` AS `Cust\_email`

   FROM

       `customers`

Select \* from cust;

Transactions

Commit

Rollback

Savepoint - is like a marker where based some condition i can switch to the savepoint

Start transaction

Insert into customers value ...

update ......

savepoint abcd;

…

…

If (condition)

rollback

Else commit

Savepoint marker

Indexes - database object

Create unique index custname on

customers(cust\_name)

Create index upper\_name on customers upper(cust\_name);

Drop index

Assignment

----------------

Create Student database studentid, student name, address, phone number, email, course

Call for ui / qa student list for which batch

Change phone number to ()- pattern

When insert student id auto where student id is

Autoincrement number + Month of join date sep + Year of join date 17

12sep17