SCS 2209 - DATABASE II

Instructions

- Download the ***.sql*** file in the UGVLE and import the file to ***phpmyadmin***.
- Use the ***salescompany*** database to write SQL queries for each question.
- Save your sql queries and result sets to a text document with the relevant question number.

  Make sure you have saved the text file using your index number.
- Then upload it to the UGVLE.

A database **index** is a data structure that improves the speed of operations in a table. Indexes are also a type of tables with primary key or index field and a pointer to each record into the actual table. The users of the database cannot see the indexes. Indexes are just used to speed up queries.

Both INSERT and UPDATE statements take more time on tables with indexes. The reason is that while doing insert or update, a database needs to insert or update the index values as well. Hence, the SELECT statements become fast on those tables.

Indexes are used to find rows with specific column values quickly. Without an index, MySQL must begin with the first row and then read through the entire table to find the relevant rows. The larger the table, the more this cost. If the table has an index for the columns in question, MySQL can quickly determine the position to seek to in the middle of the data file without having to look at all the data. This is much faster than reading every row sequentially.

Unique Index (Creating a unique index on a table)

A unique index means that two rows cannot have the same index value. We can use one or more columns to create an index. To add an index for a column or a set of columns, use the CREATE INDEX statement as follows:

Syntax :

mysql> CREATE UNIQUE INDEX index_name ON table_name (column1, column2,...);

Simple Index (Creating a simple index on a table)

A simple index allows duplicate values in a table. Omit the UNIQUE keyword from the query to create a simple index. If we need to index the values in a column in descending order, add the keyword **DESC** after the column name.ASC and DESC are also not supported for multi-valued indexes.

Syntax :

mysql> CREATE INDEX AUTHOR_INDEX ON table_name (column1 DESC);

ALTER command to add INDEX

Below statements can be used to add an index to an existing table.

Syntax :

mysql> ALTER TABLE tbl_name ADD UNIQUE index_name (column_list) ;

This statement creates a unique index for which the values must be unique (except for the NULL values, which may appear multiple times).

Syntax :

mysql> ALTER TABLE tbl_name ADD INDEX index_name (column_list) ;

This statement creates an ordinary index in which any value may appear more than once.

ALTER command to drop the *INDEX*

You can drop any INDEX by using the DROP clause along with the ALTER command.

Syntax :

mysql> ALTER TABLE tbl_name DROP INDEX index_name ;

mysql> DROP INDEX index_name ON tbl_name;

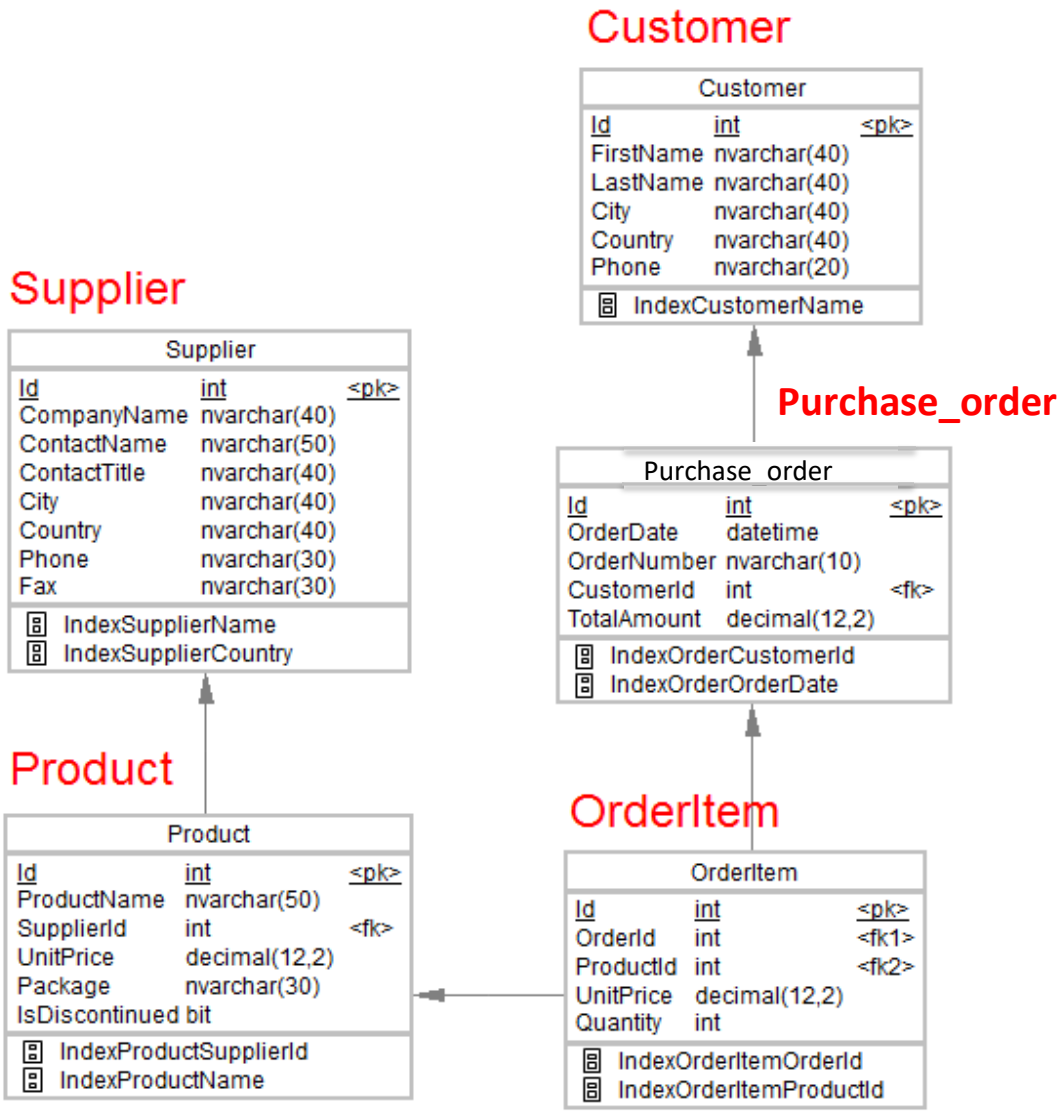
Displaying INDEX Information

You can use the SHOW INDEX command to list out all the indexes associated with a table. The vertical-format output (specified by \G) often is useful with this statement, to avoid a long line wraparound −

Syntax :

mysql> SHOW INDEXES FROM tbl_name ;

Consider the relational database for the **XYZ Sales Company** which contains details regarding their customers, suppliers, products, orders and items included in orders. The below ERD displays tables, columns, data types, relationships, primary keys, foreign keys, and indexes in the relational database.



Answer the questions below.

1. Create a note on ‘How do Indexes work in MySQL’.

2. Create an index for the column *Country* in the table *Customer* by using the CREATE INDEX statement. (view the result in structure tab)

3. Create an index for the *UnitPrice* column in *Product* table so that the unit prices will be arranged as the highest unit prices item first.

4. Show the indexes in *Customer* table.

5. Find how many customers are there whose country is “Spain”, using EXPLAIN SELECT statement.

6. Remove the index created in the question 02.

7. Identify a candidate key for customer table and create a unique index for the column.

8. *Jenny Mendel* is the daughter of *Roland Mendal*, whose Id is 20 in the *Customer* table and both of them are using the same phone number. Try to insert the following details to the customer table and show that you cannot insert duplicate values for a unique index.

Id	FirstName	LastName	City	Country	Phone
100	Jenny	Mendel	Graz	Austria	7675-3425

9. Try inserting a simple index to the same column you identified in (7) above and check whether you are able to create multiple indexes for the same column.

10. Drop all the indexes in the *Customer* table except the Primary key from a single SQL statement.