Comp2350/6350 Database Systems

Practical – Week 6

Objectives:

- To understand the use of IN operator within WHERE clause.
- To understand the use of ALIAS and CALCULATED FIELDS.
- To understand the use of Aggregate functions.
- To understand the use of GROUP BY clause.
- To understand the use of ORDER BY clause.
- To understand the use of LIMIT clause.
- To formulate and execute queries based on these operators and keywords.

Task Specifications:

Task 0: Ensuring the presence of tables with data in the database

- **You are going to work on the same database that you have created in Week 5. Ensure all tables with data exist in your database.**
- In case of any problems, please refer Task 1 of Week 5 in the appropriate document. You will find the SQL script inside Practical folder under Week 5.

Task 1: Understanding the use of IN operator within WHERE clause

- **IN** operator determines if a specified value matches any value in a list or a subquery.
- The general syntax of IN operator is:

```
SELECT  column1,column2,...
FROM    table_name
WHERE  (expr|column_1) IN ('value1','value2',...);
```

- ***Example 1:*** Find out the offices that locate in the USA and France.

```
SELECT officeCode, city, phone, country
FROM   offices
WHERE country IN ('USA', 'France');
```
[4 rows will be returned]

Task 2: Understanding the use of ALIAS and CALCULATED FIELDS

- **ALIAS** keyword can be used to give a column a descriptive name specially when calculated fields are used in a query statement.
- ***Example 2:*** Find product code, product name and the difference between their maximum retail price (MSRP) and buy price.

```
SELECT productCode, productName, (MSRP-buyPrice) AS `difference`
FROM products;
```
[110 rows will be returned]

- **The use of 'AS' keyword is optional.**

- Aliasing can also be used for table name to improve readability of multi-table query statements.
- ***Example 3:*** Find the customer names and their representative employees' first and last name.
```
SELECT customerName, lastName, firstName
FROM customers c, employees e
WHERE c.salesRepEmployeeNumber = e.employeeNumber;
```
[100 rows will be returned]

Task 3: Understanding the use of Aggregates (avg, sum, min, max, count)

- **Aggregate functions take a list of values and return exactly one value.**
- The general syntax of using aggregate functions is as follows.
```
SELECT f1(column1), f2(column2), …
FROM table_name;
```
`f1`, `f2` are aggregate functions such as avg, sum, min, max and count.
- ***Example 4:*** Find the average credit limit of customers.
```
SELECT avg(creditLimit) `Average Credit Limit` FROM customers;
```
[1 row will be returned]

Task 4: Understanding the use of GROUP BY clause

- **The `GROUP BY` clause** groups a set of rows into a set of summary rows by values of columns or expressions. The `GROUP BY` clause returns one row for each group.
- ***Example 5:*** Find the average credit limit of customers for each country.
```
SELECT country, avg(creditLimit) `Average Credit Limit`
from customers
GROUP BY country;
```
[27 rows will be returned]
- **Attributes not used by aggregate functions in the SELECT clause must appear in the GROUP BY clause.**

Task 5: Understanding the use of ORDER BY clause

- **ORDER BY** clause allows you to:
  - Sort a result set by a single column or multiple columns.
  - Sort a result set by different columns in ascending or descending order.
- The general syntax of the ORDER BY clause is:
```
SELECT   column1, column2,...
FROM     table_name
ORDER BY column1 [ASC|DESC], column2 [ASC|DESC],...;
```
- ***Example 6:*** List the contacts from the `customers` table and sorts the contacts by last name in descending order and first name in ascending order.
```
SELECT contactLastname, contactFirstname
FROM customers
ORDER BY contactLastname DESC, contactFirstname ASC;
```
[122 rows will be returned]

Task 6: Understanding the use of LIMIT clause

- **MySQL LIMIT** clause is used to constrain the number of rows returned by the `SELECT` statement.
- LIMIT clause takes two parameters.
    a. The `offset` specifies the offset of the first row to return from the result set. The `offset` of the first row is 0, not 1.
    b. The `count` specifies the maximum number of rows to return.
- ***Example 7:*** List the first 10 customers. Display only customer number, name and credit limit.

```
SELECT customernumber, customername, creditlimit
FROM customers
LIMIT 10;  [10 rows will be returned]
```

- The `LIMIT` clause often used with the `ORDER BY` clause. First, you use the `ORDER BY` clause to sort the result set based on certain criteria, and then you use the `LIMIT` clause to find lowest or highest values.
- ***Example 8:*** Find the top five customers who have the highest credit limit.

```
SELECT customernumber, customername, creditlimit
FROM customers
ORDER BY creditlimit DESC
LIMIT 5;  [5 rows will be returned]
```

Task 7: Writing and Executing Some Queries

Based on the previously introduced keywords and operators, your job is to write SQL statements for the following queries. The total number of tuples in the result set of each query is mentioned. Further a few tuples from the result set also given and highlighted for your convenience.

i) Find out the offices that locate except in the USA and France. *(Hints: use NOT IN) [3 rows will be returned]*

5	Tokyo	+81 33 224 5000	Japan
6	Sydney	+61 2 9264 2451	Australia
7	London	+44 20 7877 2041	UK

ii) Find order number, product code and the amount payable for that product (price*quantity). *(Hints: use column alias for calculated field) [1000 rows will be returned]*

orderNumber	productCode	Amount Payable
10100	S18_1749	4080.00
10100	S18_2248	2754.50
10100	S18_4409	1660.12

iii) Find order number and the total amount payable for that order (including all products). *(Hints: use appropriate aggregate function) [326 rows will be returned]*

10100	10223.83
10101	10549.01
10102	5494.78

iv) Extend the result set of the previous query by including the customer name. (Hints: use appropriate aggregate function in a multi-table query) *[326 rows will be returned]*

10100	Online Diecast Creations Co.	10223.83
10101	Blauer See Auto, Co.	10549.01
10102	Vitachrome Inc.	5494.78

v) Find order number, order date, customer name and the total amount payable of those whose customers are based in Australia. Include only the top five records based on total amount payable. *(use appropriate aggregate functions in a multi-table query with ORDER by and LIMIT clause.) [5 rows will be returned]*

10120	*2003-04-29*	*Australian Collectors, Co.*	*45864.03*
10223	*2004-02-20*	*Australian Collectors, Co.*	*44894.74*
10420	*2005-05-29*	*Souveniers And Things Co.*	*42251.51*
10347	*2004-11-29*	*Australian Collectors, Co.*	*41995.62*
10148	*2003-09-11*	*Anna's Decorations, Ltd*	*41554.73*

Appendix: Data Model of the Sample Database

This database[1] is a retailer of scale models of classic cars database. It contains typical business data such as customers, products, sales orders, sales order line items, etc. The sample database schema consists of the following tables:

- Customers: stores customer's data.
- Products: stores a list of scale model cars.
- ProductLines: stores a list of product line categories.
- Orders: stores sales orders placed by customers.
- OrderDetails: stores sales order line items for each sales order.
- Payments: stores payments made by customers based on their accounts.
- Employees: stores all employee information as well as the organization structure such as who reports to whom.
- Offices: stores sales office data.

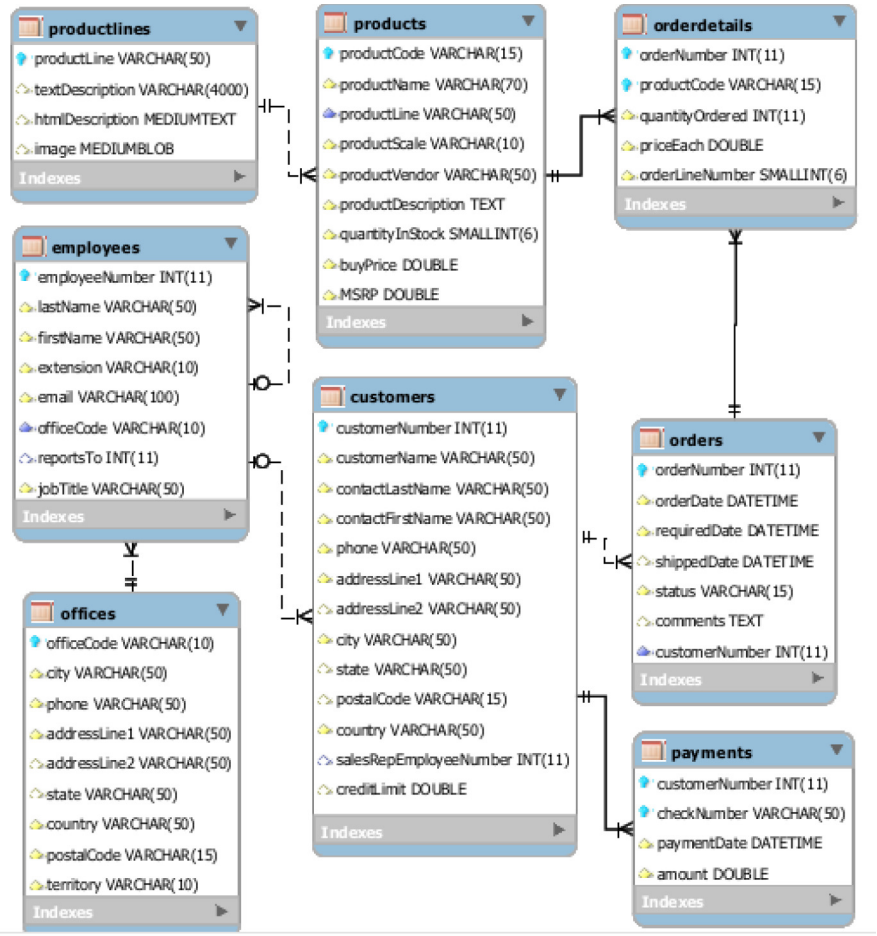

Figure 1: Data Model of Sample Database

[1] http://www.mysqltutorial.org/mysql-sample-database.aspx