**Capabilities covered:**

* Analyze complex business scenarios and create a data model
* Use MySQL to build SQL engine for an application

**Assignment 1: SQL query**

**Estimated time: 25 Mins**

**Objective:**

* How to write complex queries in MySQL

**Problem description:** Given table with the following attributes

|  |  |  |
| --- | --- | --- |
| students | | |
| STUDENT\_NO | SURNAME | FORENAME |
| 20060101 | Dickens | Charles |
| 20060102 | ApGwilym | Dafydd |
| 20060103 | Zola | Emile |
| 20060104 | Mann | Thomas |
| 20060105 | Stevenson | Robert |

|  |  |
| --- | --- |
| modules | |
| MODULE\_CODE | MODULE\_NAME |
| CM0001 | Databases |
| CM0003 | Operating Systems |
| CM0004 | Graphics |

|  |  |  |
| --- | --- | --- |
| marks | | |
| STUDENT\_NO | MODULE\_CODE | MARK |
| 20060101 | CM0001 | 80 |
| 20060101 | CM0002 | 65 |
| 20060101 | CM0003 | 50 |
| 20060102 | CM0001 | 75 |
| 20060102 | CM0003 | 45 |
| 20060102 | CM0004 | 70 |
| 20060103 | CM0001 | 60 |
| 20060103 | CM0002 | 75 |
| 20060103 | CM0004 | 60 |
| 20060104 | CM0001 | 55 |
| 20060104 | CM0002 | 40 |
| 20060104 | CM0003 | 45 |
| 20060105 | CM0001 | 55 |
| 20060105 | CM0002 | 50 |
| 20060105 | CM0004 | 65 |

Execute following statements in MySQL:

create table students ( student\_no varchar(10), surname varchar(20), forename varchar(20));

create table modules ( module\_code varchar(8), module\_name varchar(20));

create table marks ( student\_no varchar(10), module\_code varchar(8), mark integer);

insert into students values ('20060101','Dickens','Charles');

insert into students values ('20060102','ApGwilym','Dafydd');

insert into students values ('20060103','Zola','Emile');

insert into students values ('20060104','Mann','Thomas');

insert into students values ('20060105','Stevenson','Robert');

insert into modules values ('CM0001', 'Databases');

insert into modules values ('CM0002', 'Programming Languages');

insert into modules values ('CM0003', 'Operating Systems');

insert into modules values ('CM0004', 'Graphics');

insert into marks values ('20060101', 'CM0001', 80);

insert into marks values ('20060101', 'CM0002', 65);

insert into marks values ('20060101', 'CM0003', 50);

insert into marks values ('20060102', 'CM0001', 75);

insert into marks values ('20060102', 'CM0003', 45);

insert into marks values ('20060102', 'CM0004', 70);

insert into marks values ('20060103', 'CM0001', 60);

insert into marks values ('20060103', 'CM0002', 75);

insert into marks values ('20060103', 'CM0004', 60);

insert into marks values ('20060104', 'CM0001', 55);

insert into marks values ('20060104', 'CM0002', 40);

insert into marks values ('20060104', 'CM0003', 45);

insert into marks values ('20060105', 'CM0001', 55);

insert into marks values ('20060105', 'CM0002', 50);

insert into marks values ('20060105', 'CM0004', 65);

commit;

Write SQL statement for following query:

1. Increase the marks by 5 for the module operating system.

2. Update the surname of the students by adding the text “mind” before their surname. Add only the string “mind”, if it has a null value

3. Delete all the records from the database whose surname ends with the letter ‘n’

4. Fetch top 3 scores from table marks

5. Write a query which prints score of each student in each module in percentage format.

For example:

|  |  |  |
| --- | --- | --- |
| **STUDENT\_NO** | **MODULE\_CODE** | **MARK** |
| 20060101 | CM0001 | 80% |
| 20060101 | CM0002 | 65% |

Calculate the percentage based on following formula: (Marks/100)\*100

**Assignment 2: Stored procedure**

**Estimated time: 30 Mins**

**Objective:**

* How to write stored procedures in MySQL

**Problem description:** Create following table called ‘employee’

create table employee (

empId int primary key,

empName varchar(20),

empSalary int,

grade char(1)

);

Create and execute following stored procedures:

1. Create stored procedure to perform following operation EMP table:
   1. Select name from EMP table for given employee id
2. Create a stored procedure called ‘sp\_Get\_Grade’ which selects grade from EMP table for given employee id. If grade not found then assign character ‘X’ to grade.
3. Create a stored procedure called ‘sp\_Check\_Grade’ which invokes above stored procedure. If returned grade value is equal to ‘X’ then print ‘Grade not found’ else print the retrieved value
4. Create a stored procedure which increments salary of all employees by 15%
5. Write stored procedure which accepts empid and return an integer as explained below

* 1, if grade is 'M'
* 2, if grade is 'L'
* 3, if grade is 'T'
* 4, if grade is 'X'
* 0, for any exception

**Assignment 3: Triggers**

**Estimated time: 30 Mins**

**Objective:**

* How to create triggers in MySQL

**Problem description:** Create following table called ‘employee’

create table employee (

empId int primary key,

empName varchar(20),

empSalary int,

grade char(1)

);

Create and execute following stored procedures:

1. Create a trigger to be invoked before insert statement executed on employee table. It should check for grade, if grade is not matching with either one of following characters: 'M', 'L', 'T' then set it to character 'X'
2. Create following audit table which keep records of changes made to employee table:

create table employees\_audit(

id int not null auto\_increment,

empId int not null,

empName varchar(20) not null,

changedon datetime,

action varchar(50),

primary key(id)

);

Create a before update trigger to be invoked before a change is made to the employee table. This will insert a record into employee\_audit table with type of action performed on employee table**.**

**Assignment 4: SQL query**

**Estimated time: 20 Mins**

**Objective:**

* How to write complex queries in MySQL

**Problem description:**

**Refer the following tables:**

|  |  |
| --- | --- |
| **Category** | **SubCategory** |
| |  |  | | --- | --- | | **ID** | **CategoryName** | | 1 | Clothes | | 2 | Kitchen | | 3 | Sports | | 4 | Computer | | |  |  |  | | --- | --- | --- | | **SubCatID** | **SubCategoryName** | **CatID** | | 1 | ChildClothes | 1 | | 2 | Mens Wear | 1 | | 3 | CookWare | 2 | | 4 | Dinner | 2 | | 5 | Football | 3 | | 6 | Cricket | 3 | |

**Product**

|  |  |  |  |
| --- | --- | --- | --- |
| **PID** | **ProductName** | **SalesAmount** | **SubCatId** |
| 1 | winter dress | 1000 | 1 |
| 2 | summer dress | 2000 | 1 |
| 3 | night dress | 1500 | 2 |
| 4 | casuals | 3000 | 2 |
| 5 | spoons | 1000 | 3 |
| 6 | tea cups | 2500 | 3 |
| 7 | non-sticks | 5000 | 3 |
| 8 | pan | 2000 | 3 |
| 9 | Bat | 1500 | 6 |
| 10 | Ball | 2000 | 6 |

1. Write a SQL query to get the results as show below

Total Sales of a product with result columns as (SubCategoryName,TotalSales)]

|  |  |
| --- | --- |
| **SubCategoryName** | **TotalSales** |
| ChildClothes | 3000 |
| Men's Wear | 4500 |
| CookWare | 10500 |
| Cricket | 3500 |
| Dinner | 0 |
| Football | 0 |

1. Write a SQL query to get the results as show below

[Total sales of a category with result column as (CategoryName,TotalSales)]

|  |  |
| --- | --- |
| **CategoryName** | **TotalSales** |
| Clothes | 7500 |
| Kitchen | 10500 |
| Sports | 3500 |
| Computer | 0 |

**Assignment 5: SQL query**

**Estimated time: 20 Mins**

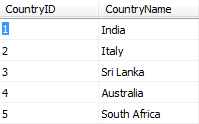
**Objective:**

* How to write complex queries in MySQL

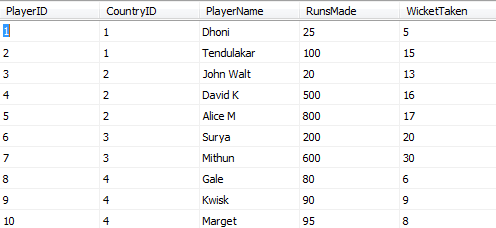
**Problem description:**

Refer the following tables:

**Country table**



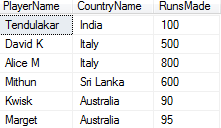
**Player table**



**Description:**

Select the player name, country the player belongs to and runs made, where the runs made by the player is more than average runs of all players within his country

Refer the following output:



**Assignment 6: SQL query**

**Estimated time: 30 Mins**

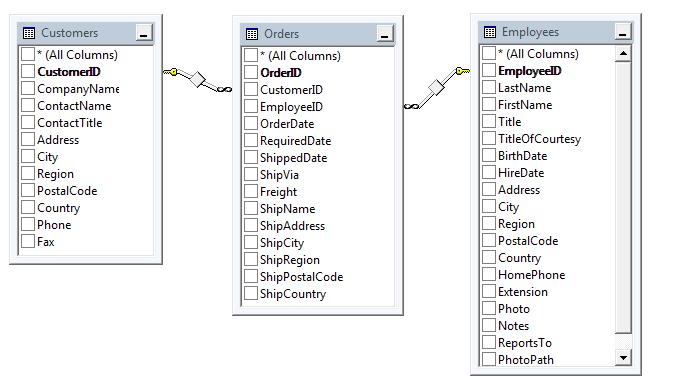
**Objective:**

* How to write complex queries in MySQL

**Problem description:**

Create a report showing the Order ID, the name of the company that placed the order, and the first and last name of the associated employee. Only show orders placed after January 1, 1998 that shipped after they were required. Sort by Company Name.

**The structure of the tables is as follows:**



**The output should be as follows:**



**Summary of assignments:**

You have learnt:

* How to write complex queries in MySQL?
* How to write stored procedures?
* How to process table data inside stored procedures?
* How to write triggers in MySQL?
* How to fire triggers during various events?