**MS SQL Assignment**

**Name:**

**Note: Read the following carefully**

* Any late submission will be considered as a grade of 0%

### I do not accept your report or sql query through email.

### Only one person on be half of the group submit

* **Upload your sql file only.**
* **Name it as Group number. For example, if you are in in group 2 name your sql file as G2**
* **Do not put any name in the sql file.**
* **Below each question provide your query**

**Mini project – Student Data Management**

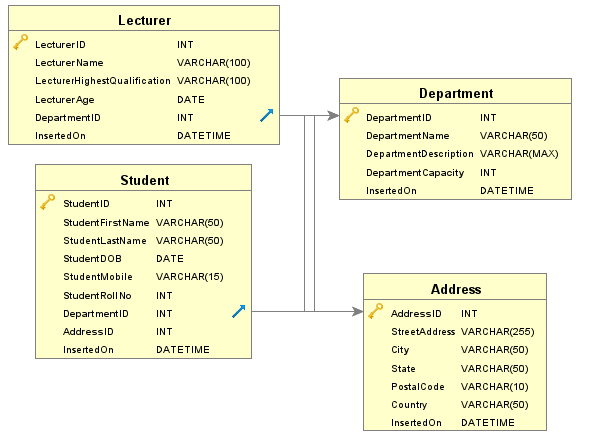
1. Create a database called “LMT\_University”.

**CREATE DATABASE LMT\_University;**

1. Create an “enrol” schema under “LMT\_University”.

**CREATE SCHEMA enrol**;

1. Use “enrol” Schema for creating the project.
2. Create the Database modeling in such a way that Model looks like-



1. Create an “Address” table under “enrol” schema with the following specifications-
   1. Address table must have the following attributes-

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| AddressID | Integer | Address Unique Identifier |
| StreetAddress | Varchar | Street Address |
| City | Varchar | City |
| State | Varchar | State |
| PostalCode | Varchar | Zip Code of a particular location |
| Country | Varchar | Country |
| InsertedOn | DateTime | Data Insertion Date |

* 1. Add the following constraint.
     1. AddressID: Auto Increment, Primary Key, Not null, clustered index.
     2. StreetAddress: Null
     3. City: Not Null
     4. State: Null
     5. PostalCode: Null
     6. Country: Not Null
     7. InsertedOn: Not Null

**CREATE TABLE enrol.Address (**

**AddressID INT IDENTITY(1,1) NOT NULL PRIMARY KEY CLUSTERED,**

**StreetAddress nvarchar(50) NOT NULL,**

**City nvarchar(50) NOT NULL,**

**State nvarchar(50) NULL,**

**PostalCode nvarchar(50) NULL,**

**Country nvarchar(50) NULL,**

**InsertedOn Date NOT NULL DEFAULT GETDATE()**

**);**

* 1. Insert the following records based on the following specifications.
     1. AddressID: Address ID starting from 1 to 30 with step size 1.
     2. StreetAddress: Insert the street Address mentioned in the table.
     3. City: Insert the City mentioned in the table.
     4. State: Insert the State mentioned in the table.
     5. PostalCode: Insert the PostalCode mentioned in the table.
     6. Country: Insert the Country mentioned in the table.
     7. InsertedOn: Insert as default date as system date.
  2. After Insertion, Table looks like as shown below-

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **AddressID** |  | **StreetAddress** | **City** | **State** | **PostalCode** | **Country** | **InsertedOn** |
| 1 |  | 5 Schurz Lane | Grybów | NULL | 33-330 | Poland | 2020-09-30 |
| 2 |  | 628 Waubesa Drive | Jinsheng | NULL | NULL | China | 2020-09-30 |
| 3 |  | 44135 Northfield Way | Nowy Dwór Mazowiecki | NULL | 05-160 | Poland | 2020-09-30 |
| 4 |  | 335 Bellgrove Road | Gaoqiao | NULL | NULL | China | 2020-09-30 |
| 5 |  | 28 Victoria Junction | Bukovec | NULL | 739 84 | Czech Republic | 2020-09-30 |
| 6 |  | 6 Stuart Road | Wushan | NULL | NULL | China | 2020-09-30 |
| 7 |  | 730 Barby Street | Zhengchang | NULL | NULL | China | 2020-09-30 |
| 8 |  | 22742 Schiller Street | Sumurwaru | NULL | NULL | Indonesia | 2020-09-30 |
| 9 |  | 31 Elka Junction | Cigembong | NULL | NULL | Indonesia | 2020-09-30 |
| 10 |  | 5 Kenwood Circle | Davao | NULL | 8000 | Philippines | 2020-09-30 |
| 11 |  | 99 Bunker Hill Crossing | Zarasai | NULL | 32001 | Lithuania | 2020-09-30 |
| 12 |  | 5 Farragut Center | Jaromerice | NULL | 569 44 | Czech Republic | 2020-09-30 |
| 13 |  | 25 Lerdahl Street | Nanshi | NULL | NULL | China | 2020-09-30 |
| 14 |  | 918 Bonner Way | Phayakkhaphum Phisai | NULL | 44110 | Thailand | 2020-09-30 |
| 15 |  | 9 West Alley | Sempu | NULL | NULL | Indonesia | 2020-09-30 |
| 16 |  | 234 Hagan Lane | Rennes | Bretagne | 35033 | France | 2020-09-30 |
| 17 |  | 33942 Eagle Crest Trail | Oliveiras | Porto | 4745-235 | Portugal | 2020-09-30 |
| 18 |  | 20791 Hermina Way | B?o L?c | NULL | NULL | Vietnam | 2020-09-30 |
| 19 |  | 86 Lake View Way | Marsa Alam | NULL | NULL | Egypt | 2020-09-30 |
| 20 |  | 19732 Burning Wood Parkway | Piteå | Norrbotten | 944 73 | Sweden | 2020-09-30 |
| 21 |  | 9320 Oak Valley Road | Rathangani | NULL | A45 | Ireland | 2020-09-30 |
| 22 |  | 2638 Waubesa Circle | Honda | NULL | 732048 | Colombia | 2020-09-30 |
| 23 |  | 6999 Monument Center | Cortes | NULL | 6341 | Philippines | 2020-09-30 |
| 24 |  | 1 Warbler Hill | Proletar | NULL | NULL | Tajikistan | 2020-09-30 |
| 25 |  | 1311 Crowley Street | Baghlan | NULL | NULL | Afghanistan | 2020-09-30 |
| 26 |  | 19 Walton Way | Öldziyt | NULL | NULL | Mongolia | 2020-09-30 |
| 27 |  | 1 Glacier Hill | Cergy-Pontoise | Île-de-France | 95304 | France | 2020-09-30 |
| 28 |  | 5094 Gateway Way | Živinice | NULL | NULL | Bosnia and Herzegovina | 2020-09-30 |
| 29 |  | 2 Roth Pass | Tuatuka | NULL | NULL | Indonesia | 2020-09-30 |
| 30 |  | 89531 Northview Road | Ganyi | NULL | NULL | China | 2020-09-30 |

**INSERT INTO enrol.Address(StreetAddress,City,State,PostalCode,Country) VALUES ('5 Schurz Lane','Grybow',NULL,'33-330','Poland');**

**INSERT INTO enrol.Address(StreetAddress,City,State,PostalCode,Country) VALUES ('628 Waubesa Drive','Jinsheng',NULL,NULL,'China');**

**INSERT INTO enrol.Address(StreetAddress,City,State,PostalCode,Country) VALUES ('44135 Northfield Way','Nowy Dwor Mazowiecki',NULL,'05-160','Poland');**

**INSERT INTO enrol.Address(StreetAddress,City,State,PostalCode,Country) VALUES ('335 Bellgrove Road','Gaoqiao',NULL,NULL,'China');**

**INSERT INTO enrol.Address(StreetAddress,City,State,PostalCode,Country) VALUES ('28 Victoria Junction','Bukovec',NULL,'739 84','Czech Republic');**

**INSERT INTO enrol.Address(StreetAddress,City,State,PostalCode,Country) VALUES ('6 Stuart Road','Wushan',NULL,NULL,'China');**

**INSERT INTO enrol.Address(StreetAddress,City,State,PostalCode,Country) VALUES ('730 Barby Street','Zhengchang',NULL,NULL,'China');**

**INSERT INTO enrol.Address(StreetAddress,City,State,PostalCode,Country) VALUES ('22742 Schiller Street','Sumurwaru',NULL,NULL,'Indonesia');**

**INSERT INTO enrol.Address(StreetAddress,City,State,PostalCode,Country) VALUES ('31 Elka Junction','Cigembong',NULL,NULL,'Indonesia');**

**INSERT INTO enrol.Address(StreetAddress,City,State,PostalCode,Country) VALUES ('5 Kenwood Circle','Davao',NULL,'8000','Philippines');**

**INSERT INTO enrol.Address(StreetAddress,City,State,PostalCode,Country) VALUES ('99 Bunker Hill Crossing','Zarasai',NULL,'32001','Lithuania');**

**INSERT INTO enrol.Address(StreetAddress,City,State,PostalCode,Country) VALUES ('5 Farragut Center','Jaromerice',NULL,'569 44','Czech Republic');**

**INSERT INTO enrol.Address(StreetAddress,City,State,PostalCode,Country) VALUES ('25 Lerdahl Street','Nanshi',NULL,NULL,'China');**

**INSERT INTO enrol.Address(StreetAddress,City,State,PostalCode,Country) VALUES ('918 Bonner Way','Phayakkhaphum Phisai',NULL,'44110','Thailand');**

**INSERT INTO enrol.Address(StreetAddress,City,State,PostalCode,Country) VALUES ('9 West Alley','Sempu',NULL,NULL,'Indonesia');**

**INSERT INTO enrol.Address(StreetAddress,City,State,PostalCode,Country) VALUES ('234 Hagan Lane','Rennes','Bretagne','35033','France');**

**INSERT INTO enrol.Address(StreetAddress,City,State,PostalCode,Country) VALUES ('33942 Eagle Crest Trail','Oliveiras','Porto','4745-235','Portugal');**

**INSERT INTO enrol.Address(StreetAddress,City,State,PostalCode,Country) VALUES ('20791 Hermina Way','B?o L?c',NULL,NULL,'Vietnam');**

**INSERT INTO enrol.Address(StreetAddress,City,State,PostalCode,Country) VALUES ('86 Lake View Way','Marsa Alam',NULL,NULL,'Egypt');**

**INSERT INTO enrol.Address(StreetAddress,City,State,PostalCode,Country) VALUES ('19732 Burning Wood Parkway','Pitea','Norrbotten','944 73','Sweden');**

**INSERT INTO enrol.Address(StreetAddress,City,State,PostalCode,Country) VALUES ('9320 Oak Valley Road','Rathangani',NULL,'A45','Ireland');**

**INSERT INTO enrol.Address(StreetAddress,City,State,PostalCode,Country) VALUES ('2638 Waubesa Circle','Honda',NULL,'732048','Colombia');**

**INSERT INTO enrol.Address(StreetAddress,City,State,PostalCode,Country) VALUES ('6999 Monument Center','Cortes',NULL,'6341','Philippines');**

**INSERT INTO enrol.Address(StreetAddress,City,State,PostalCode,Country) VALUES ('1 Warbler Hill','Proletar',NULL,NULL,'Tajikistan');**

**INSERT INTO enrol.Address(StreetAddress,City,State,PostalCode,Country) VALUES ('1311 Crowley Street','Baghlan',NULL,NULL,'Afghanistan');**

**INSERT INTO enrol.Address(StreetAddress,City,State,PostalCode,Country) VALUES ('19 Walton Way','aldziyt',NULL,NULL,'Mongolia');**

**INSERT INTO enrol.Address(StreetAddress,City,State,PostalCode,Country) VALUES ('1 Glacier Hill','Cergy-Pontoise','ile-de-France','95304','France');**

**INSERT INTO enrol.Address(StreetAddress,City,State,PostalCode,Country) VALUES ('5094 Gateway Way','zivinice',NULL,NULL,'Bosnia and Herzegovina');**

**INSERT INTO enrol.Address(StreetAddress,City,State,PostalCode,Country) VALUES ('2 Roth Pass','Tuatuka',NULL,NULL,'Indonesia');**

**INSERT INTO enrol.Address(StreetAddress,City,State,PostalCode,Country) VALUES ('89531 Northview Road','Ganyi',NULL,NULL,'China');**

1. Create a “Department” table under “enrol” schema with the following specifications-

**CREATE TABLE enrol.Department (**

**DepartmentID INT IDENTITY(1,1) NOT NULL PRIMARY KEY CLUSTERED,**

**DepartmentName nvarchar(50) NOT NULL,**

**DepartmentDescription nvarchar(500) NULL,**

**DepartmentCapacity INT NOT NULL,**

**InsertedOn Date NOT NULL DEFAULT GETDATE()**

**);**

* 1. Department table Must have the following attributes-

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| DepartmentID | Integer | Department Unique Identifier |
| DepartmentName | Lecturer | Department Name |
| DepartmentDescription | Varchar | Department Description |
| DepartmentCapacity | Integer | Department Maximum Occupancy |
| InsertedOn | DateTime | Data Insertion Date |

* 1. Add the following constraints-
     1. DepartmentID: Auto Increment, Primary Key, Not null, clustered index.
     2. DepartmentName: Not Null
     3. DepartmentDescription: Null
     4. DepartmentCapacity: Not Null
     5. InsertedOn: Not Null
  2. Insert the following records based on the following specifications.
     1. DepartmentID: Department ID starting from 1 to 30 with step size 1.
     2. DepartmentName: Insert the Department Name mentioned in the table.
     3. DepartmentDescription: Insert the Department Description mentioned in the table.
     4. DepartmentCapacity: Insert the Department Capacity mentioned in the table.
     5. InsertedOn: Insert as default date as system date.

**INSERT INTO enrol.Department(DepartmentName,DepartmentDescription,DepartmentCapacity,InsertedOn) VALUES ('IT','Information Technology',60,'9/30/2020');**

**INSERT INTO enrol.Department(DepartmentName,DepartmentDescription,DepartmentCapacity,InsertedOn) VALUES ('EE','Electrical Engineering',120,'9/30/2020');**

**INSERT INTO enrol.Department(DepartmentName,DepartmentDescription,DepartmentCapacity,InsertedOn) VALUES ('CSE','Computer Science Engineering',140,'9/30/2020');**

**INSERT INTO enrol.Department(DepartmentName,DepartmentDescription,DepartmentCapacity,InsertedOn) VALUES ('ME','Mechanical Engineering',110,'9/30/2020');**

**INSERT INTO enrol.Department(DepartmentName,DepartmentDescription,DepartmentCapacity,InsertedOn) VALUES ('ECE','Electronic and Communication Engineering',80,'9/30/2020');**

**INSERT INTO enrol.Department(DepartmentName,DepartmentDescription,DepartmentCapacity,InsertedOn) VALUES ('AEIE','Applied Electronics and Instrumentation Engineering',50,'9/30/2020');**

* 1. After Insertion, Table looks like as shown below-

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DepartmentID** | **DepartmentName** | **DepartmentDescription** | **DepartmentCapacity** | **InsertedOn** |
| 1 | IT | Information Technology | 60 | 2020-09-30 |
| 2 | EE | Electrical Engineering | 120 | 2020-09-30 |
| 3 | CSE | Computer Science Engineering | 140 | 2020-09-30 |
| 4 | ME | Mechanical Engineering | 110 | 2020-09-30 |
| 5 | ECE | Electronic and Communication Engineering | 80 | 2020-09-30 |
| 6 | AEIE | Applied Electronics and Instrumentation Engineering | 50 | 2020-09-30 |

1. Create a “Lecturer” table under “enrol” schema with the following specifications-

**CREATE TABLE enrol.Lecturer**

**(**

**LecturerID INT IDENTITY(1,1) NOT NULL PRIMARY KEY CLUSTERED,**

**LecturerName nvarchar(50) NOT NULL,**

**LecturerHighestQualification nvarchar(50) NULL,**

**LecturerAge nvarchar(50) NOT NULL,**

**DepartmentID INT NOT NULL,**

**CONSTRAINT FK\_Lectures\_Department FOREIGN KEY (DepartmentID)**

**REFERENCES enrol.Department (DepartmentID)**

**ON DELETE CASCADE**

**ON UPDATE CASCADE,**

**InsertedOn Date NULL DEFAULT GETDATE()**

**)**

**;**

* 1. Lecturer table Must have the following attributes-

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| LecturerID | Integer | Lecturer Unique Identifier |
| LecturerName | Varchar | Lecturer Name |
| LecturerHighestQualification | Varchar | Lecturer Highest Qualification |
| LecturerAge | Varchar | Lecturer Joining Date |
| DepartmentID | Integer | Department Unique Identifier |
| InsertedOn | DateTime | Data Insertion Date |

* 1. Add the following constraints-
     1. LecturerID: Auto Increment, Primary Key, Not null, clustered index.
     2. LecturerName: Not Null
     3. LecturerHighestQualification: Null
     4. LecturerAge: Not Null
     5. DepartmentID: Not Null
     6. InsertedOn: Null
  2. Insert the following records based on the following specifications.
     1. LecturerID: Lecturer ID starting from 1 to 30 with step size 1.
     2. LecturerName: Insert the Lecturer Name mentioned in the table.
     3. LecturerHighestQualification: Insert the Lecturer Highest Qualification mentioned in the table.
     4. LecturerAge: Insert the Lecturer Age mentioned in the table.
     5. DepartmentID: Foreign key, Not Null.
     6. InsertedOn: Insert as default date as system date.
  3. After Insertion, Table looks like as shown below-

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **LecturerID** | **LecturerName** | **LecturerHighestQualification** | **LecturerAge** | **DepartmentID** | **InsertedOn** |
| 1 | Peder Bernaldez | M.Tech | 2010-10-10 | 6 | 2020-09-30 |
| 2 | Emile Adolthine | PhD | 2010-04-04 | 5 | 2020-09-30 |
| 3 | Titos Iorizzi | M.Tech | 2012-04-09 | 4 | 2020-09-30 |
| 4 | Ferris Falck | MSC | 2011-05-05 | 3 | 2020-09-30 |
| 5 | Georgie McIlwraith | M.Tech | 2017-05-08 | 2 | 2020-09-30 |
| 6 | Karlen Kearn | MSC | 2019-03-03 | 1 | 2020-09-30 |
| 7 | Axe Whistlecroft | MCA | 2019-03-03 | 6 | 2020-09-30 |
| 8 | Drucie Bazek | PhD | 2019-04-01 | 5 | 2020-09-30 |
| 9 | Antony Gamlin | M.Tech | 2019-04-01 | 4 | 2020-09-30 |
| 10 | Alexina Moncaster | MBA | 2019-04-01 | 3 | 2020-09-30 |
| 11 | Milzie Kabos | MCA | 2019-03-03 | 2 | 2020-09-30 |
| 12 | Arlene Glendza | MS | 2019-03-03 | 1 | 2020-09-30 |
| 13 | Kirby Kabisch | M.Tech | 2019-04-01 | 1 | 2020-09-30 |
| 14 | Selma Eliyahu | PhD | 2019-04-01 | 2 | 2020-09-30 |
| 15 | Ilysa Chooter | M.Tech | 2019-04-01 | 3 | 2020-09-30 |
| 16 | Rozalie Pennycord | MSC | 2010-10-10 | 4 | 2020-09-30 |
| 17 | Dacey Glidder | M.Tech | 2010-04-04 | 5 | 2020-09-30 |
| 18 | Claretta Diaper | MSC | 2012-04-09 | 6 | 2020-09-30 |
| 19 | Kalil Pendleton | MCA | 2011-05-05 | 6 | 2020-09-30 |
| 20 | Trudey Brech | PhD | 2011-10-05 | 5 | 2020-09-30 |
| 21 | Gypsy Ambrosini | M.Tech | 2011-03-30 | 4 | 2020-09-30 |
| 22 | Lauree Ribbon | MBA | 2013-04-04 | 3 | 2020-09-30 |
| 23 | Hugo Valois | MCA | 2012-04-29 | 2 | 2020-09-30 |
| 24 | Perren Chetter | MS | 2018-05-03 | 1 | 2020-09-30 |
| 25 | Fawn Coffelt | M.Tech | 2020-02-26 | 1 | 2020-09-30 |
| 26 | Terrie Golby | PhD | 2020-02-26 | 2 | 2020-09-30 |
| 27 | Jeanette Ciraldo | M.Tech | 2020-03-26 | 3 | 2020-09-30 |
| 28 | Elfrieda Elijahu | MSC | 2020-03-26 | 4 | 2020-09-30 |
| 29 | Guthry Blaes | M.Tech | 2020-03-26 | 5 | 2020-09-30 |
| 30 | Richy Saice | MSC | 2020-02-26 | 6 | 2020-09-30 |

**INSERT INTO enrol.Lecturer(LecturerName,LecturerHighestQualification,LecturerAge,DepartmentID) VALUES ('Peder Bernaldez','M.Tech','10/10/2010',6);**

**INSERT INTO enrol.Lecturer(LecturerName,LecturerHighestQualification,LecturerAge,DepartmentID) VALUES ('Emile Adolthine','PhD','4/4/2010',5);**

**INSERT INTO enrol.Lecturer(LecturerName,LecturerHighestQualification,LecturerAge,DepartmentID) VALUES ('Titos Iorizzi','M.Tech','4/9/2012',4);**

**INSERT INTO enrol.Lecturer(LecturerName,LecturerHighestQualification,LecturerAge,DepartmentID) VALUES ('Ferris Falck','MSC','5/5/2011',3);**

**INSERT INTO enrol.Lecturer(LecturerName,LecturerHighestQualification,LecturerAge,DepartmentID) VALUES ('Georgie McIlwraith','M.Tech','5/8/2017',2);**

**INSERT INTO enrol.Lecturer(LecturerName,LecturerHighestQualification,LecturerAge,DepartmentID) VALUES ('Karlen Kearn','MSC','3/3/2019',1);**

**INSERT INTO enrol.Lecturer(LecturerName,LecturerHighestQualification,LecturerAge,DepartmentID) VALUES ('Axe Whistlecroft','MCA','3/3/2019',6);**

**INSERT INTO enrol.Lecturer(LecturerName,LecturerHighestQualification,LecturerAge,DepartmentID) VALUES ('Drucie Bazek','PhD','4/1/2019',5);**

**INSERT INTO enrol.Lecturer(LecturerName,LecturerHighestQualification,LecturerAge,DepartmentID) VALUES ('Antony Gamlin','M.Tech','4/1/2019',4);**

**INSERT INTO enrol.Lecturer(LecturerName,LecturerHighestQualification,LecturerAge,DepartmentID) VALUES ('Alexina Moncaster','MBA','4/1/2019',3);**

**INSERT INTO enrol.Lecturer(LecturerName,LecturerHighestQualification,LecturerAge,DepartmentID) VALUES ('Milzie Kabos','MCA','3/3/2019',2);**

**INSERT INTO enrol.Lecturer(LecturerName,LecturerHighestQualification,LecturerAge,DepartmentID) VALUES ('Arlene Glendza','MS','3/3/2019',1);**

**INSERT INTO enrol.Lecturer(LecturerName,LecturerHighestQualification,LecturerAge,DepartmentID) VALUES ('Kirby Kabisch','M.Tech','4/1/2019',1);**

**INSERT INTO enrol.Lecturer(LecturerName,LecturerHighestQualification,LecturerAge,DepartmentID) VALUES ('Selma Eliyahu','PhD','4/1/2019',2);**

**INSERT INTO enrol.Lecturer(LecturerName,LecturerHighestQualification,LecturerAge,DepartmentID) VALUES ('Ilysa Chooter','M.Tech','4/1/2019',3);**

**INSERT INTO enrol.Lecturer(LecturerName,LecturerHighestQualification,LecturerAge,DepartmentID) VALUES ('Rozalie Pennycord','MSC','10/10/2010',4);**

**INSERT INTO enrol.Lecturer(LecturerName,LecturerHighestQualification,LecturerAge,DepartmentID) VALUES ('Dacey Glidder','M.Tech','4/4/2010',5);**

**INSERT INTO enrol.Lecturer(LecturerName,LecturerHighestQualification,LecturerAge,DepartmentID) VALUES ('Claretta Diaper','MSC','4/9/2012',6);**

**INSERT INTO enrol.Lecturer(LecturerName,LecturerHighestQualification,LecturerAge,DepartmentID) VALUES ('Kalil Pendleton','MCA','5/5/2011',6);**

**INSERT INTO enrol.Lecturer(LecturerName,LecturerHighestQualification,LecturerAge,DepartmentID) VALUES ('Trudey Brech','PhD','10/5/2011',5);**

**INSERT INTO enrol.Lecturer(LecturerName,LecturerHighestQualification,LecturerAge,DepartmentID) VALUES ('Gypsy Ambrosini','M.Tech','3/30/2011',4);**

**INSERT INTO enrol.Lecturer(LecturerName,LecturerHighestQualification,LecturerAge,DepartmentID) VALUES ('Lauree Ribbon','MBA','4/4/2013',3);**

**INSERT INTO enrol.Lecturer(LecturerName,LecturerHighestQualification,LecturerAge,DepartmentID) VALUES ('Hugo Valois','MCA','4/29/2012',2);**

**INSERT INTO enrol.Lecturer(LecturerName,LecturerHighestQualification,LecturerAge,DepartmentID) VALUES ('Perren Chetter','MS','5/3/2018',1);**

**INSERT INTO enrol.Lecturer(LecturerName,LecturerHighestQualification,LecturerAge,DepartmentID) VALUES ('Fawn Coffelt','M.Tech','2/26/2020',1);**

**INSERT INTO enrol.Lecturer(LecturerName,LecturerHighestQualification,LecturerAge,DepartmentID) VALUES ('Terrie Golby','PhD','2/26/2020',2);**

**INSERT INTO enrol.Lecturer(LecturerName,LecturerHighestQualification,LecturerAge,DepartmentID) VALUES ('Jeanette Ciraldo','M.Tech','3/26/2020',3);**

**INSERT INTO enrol.Lecturer(LecturerName,LecturerHighestQualification,LecturerAge,DepartmentID) VALUES ('Elfrieda Elijahu','MSC','3/26/2020',4);**

**INSERT INTO enrol.Lecturer(LecturerName,LecturerHighestQualification,LecturerAge,DepartmentID) VALUES ('Guthry Blaes','M.Tech','3/26/2020',5);**

**INSERT INTO enrol.Lecturer(LecturerName,LecturerHighestQualification,LecturerAge,DepartmentID) VALUES ('Richy Saice','MSC','2/26/2020',6);**

1. Create a “Student” table under “enrol” schema with the following specifications-

**CREATE TABLE enrol.Student**

**(**

**StudentID INT IDENTITY(1,1) NOT NULL PRIMARY KEY CLUSTERED,**

**StudentFirstName nvarchar(50) NOT NULL,**

**StudentLastName nvarchar(50) NULL,**

**StudentDOB Date NOT NULL,**

**StudentMobile nvarchar(50) NULL,**

**StudentRollNo INT NOT NULL,**

**DepartmentID INT NOT NULL,**

**CONSTRAINT FK\_Student\_Department FOREIGN KEY (DepartmentID)**

**REFERENCES enrol.Department (DepartmentID)**

**ON DELETE CASCADE**

**ON UPDATE CASCADE,**

**AddressID INT NOT NULL,**

**CONSTRAINT FK\_Student\_Address FOREIGN KEY (AddressID)**

**REFERENCES enrol.Address (AddressID)**

**ON DELETE CASCADE**

**ON UPDATE CASCADE,**

**InsertedOn Date NOT NULL DEFAULT GETDATE()**

**)**

**;**

* 1. Student table Must have the following attributes-

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| StudentID | Integer | Student Unique Identifier |
| StudentFirstName | Varchar | Student First Name |
| StudentLastName | Varchar | Student Last Name |
| StudentDOB | Date | Student Date of Birth |
| StudentMobile | Varchar | Department Unique Identifier |
| StudentRollNo | Integer | Student Roll Number |
| DepartmentID | Integer | Department Unique Identifier |
| AddressID | Integer | Address Unique Identifier |
| InsertedOn | DateTime | Data Insertion Date |

* 1. Add the following constraints-
     1. StudentID: Auto Increment, Primary Key, Not null, clustered index.
     2. StudentFirstName: Not Null
     3. StudentLastName: Null
     4. StudentDOB: Not Null
     5. StudentMobile: Null
     6. StudentRollNo: Not Null
     7. DepartmentID: Foreign key, Not Null.
     8. AddressID: Foreign key, Not Null.
     9. InsertedOn: Not Null
  2. Insert the following records based on the following specifications.
     1. StudentID: Student ID starting from 1 to 50 with step size 1.
     2. StudentFirstName: Insert the Student First Name mentioned in the table.
     3. StudentLastName: Insert the Student Last Name mentioned in the table.
     4. StudentDOB: Insert the Student DOB mentioned in the table.
     5. StudentMobile: Insert the Student Mobile no mentioned in the table.
     6. StudentRollNo: Insert the Student Roll no mentioned in the table.
     7. DepartmentID: Insert the Department ID mentioned in the table.
     8. AddressID: Insert the Address ID mentioned in the table.
     9. InsertedOn: Insert as default date as system date.
  3. After Insertion, Table looks like as shown below-

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **StudentID** | **StudentFirstName** | **StudentLastName** | **StudentDOB** | **StudentMobile** | **StudentRollNo** | **DepartmentID** | **AddressID** | **InsertedOn** |
| 1 | Joey | Ironside | 1995-11-22 | 1276234258 | 1 | 3 | 1 | 2020-10-01 |
| 2 | Karlotta | Garraway | 1997-07-06 | 2192431615 | 2 | 3 | 24 | 2020-10-01 |
| 3 | Jerry | Stutte | 1996-12-18 | 4125425783 | 3 | 1 | 17 | 2020-10-01 |
| 4 | Yehudit | Rahill | 1995-01-15 | 9939485406 | 4 | 2 | 29 | 2020-10-01 |
| 5 | Cele | Crosetto | 1998-11-24 | 3622733725 | 5 | 3 | 16 | 2020-10-01 |
| 6 | Hazlett | Mowsdale | 1995-04-09 | 1482883476 | 6 | 4 | 23 | 2020-10-01 |
| 7 | Carlyn | Marks | 1996-12-27 | 6129154080 | 7 | 5 | 20 | 2020-10-01 |
| 8 | Ellis | Boatman | 1997-04-29 | 8269707118 | 8 | 6 | 7 | 2020-10-01 |
| 9 | Florina | Boyack | 1997-08-03 | 9623352863 | 9 | 3 | 14 | 2020-10-01 |
| 10 | Borg | Innett | 1997-09-03 | 5256034960 | 10 | 1 | 19 | 2020-10-01 |
| 11 | Sayres | Jennings | 1996-05-12 | 8675076454 | 11 | 4 | 27 | 2020-10-01 |
| 12 | Jarid | Sprull | 1998-11-02 | 1391270091 | 12 | 2 | 6 | 2020-10-01 |
| 13 | Elvera | Bannard | 1996-09-07 | 7897232539 | 13 | 4 | 24 | 2020-10-01 |
| 14 | Ody | Inggall | 1995-03-05 | 6094734260 | 14 | 5 | 25 | 2020-10-01 |
| 15 | Curcio | McWhan | 1996-07-29 | 2394865847 | 15 | 6 | 11 | 2020-10-01 |
| 16 | Connie | Sinnie | 1995-07-19 | 1473936221 | 16 | 6 | 23 | 2020-10-01 |
| 17 | Auroora | Nel | 1996-09-05 | 2216400391 | 17 | 3 | 14 | 2020-10-01 |
| 18 | Wendall | Rosendale | 1999-12-30 | 1818120249 | 18 | 3 | 28 | 2020-10-01 |
| 19 | Hadley | Bradbury | 1996-08-16 | 6518067697 | 19 | 1 | 10 | 2020-10-01 |
| 20 | Celine | Smales | 1999-07-11 | 7106508130 | 20 | 2 | 10 | 2020-10-01 |
| 21 | Jesselyn | Stevenson | 1998-05-16 | 9231672206 | 21 | 2 | 22 | 2020-10-01 |
| 22 | Corinna | Pinkney | 1998-01-16 | 8323630067 | 22 | 5 | 29 | 2020-10-01 |
| 23 | Orelle | Adamthwaite | 1997-07-26 | 2539126766 | 23 | 3 | 17 | 2020-10-01 |
| 24 | Howie | Seaman | 1997-12-01 | 9888259627 | 24 | 2 | 4 | 2020-10-01 |
| 25 | Sibyl | Corey | 1996-07-18 | 4493239590 | 25 | 5 | 11 | 2020-10-01 |
| 26 | Ruperta | Peaker | 1999-05-22 | 5124781263 | 26 | 5 | 4 | 2020-10-01 |
| 27 | Delmer | Roughey | 1995-04-21 | 4175314364 | 27 | 3 | 22 | 2020-10-01 |
| 28 | Gifford | O'Scannill | 1996-10-31 | 3134783726 | 28 | 4 | 22 | 2020-10-01 |
| 29 | Hedy | O'Hone | 1998-03-29 | 7316228047 | 29 | 2 | 17 | 2020-10-01 |
| 30 | Shalna | Hyde-Chambers | 1999-11-23 | 7455116160 | 30 | 5 | 6 | 2020-10-01 |
| 31 | Ferdie | Di Napoli | 1995-01-17 | 1905908693 | 31 | 4 | 30 | 2020-10-01 |
| 32 | Piper | Giacomuzzo | 1998-09-14 | 5499340503 | 32 | 6 | 4 | 2020-10-01 |
| 33 | Gerhardt | Schruurs | 1999-11-18 | 8197494894 | 33 | 3 | 1 | 2020-10-01 |
| 34 | Mellicent | Buncher | 1996-10-03 | 4584525312 | 34 | 5 | 28 | 2020-10-01 |
| 35 | Corette | Demead | 1997-09-17 | 4909862137 | 35 | 5 | 17 | 2020-10-01 |
| 36 | Jorgan | Barson | 1997-05-01 | 6022309183 | 36 | 1 | 21 | 2020-10-01 |
| 37 | Koral | Bowen | 1998-05-12 | 4198817454 | 37 | 4 | 3 | 2020-10-01 |
| 38 | Allissa | Kitter | 1998-08-17 | 7328676920 | 38 | 5 | 7 | 2020-10-01 |
| 39 | Townsend | Doughtery | 1998-04-13 | 2639777958 | 39 | 4 | 7 | 2020-10-01 |
| 40 | Yolane | Geratt | 1998-06-10 | 2069585951 | 40 | 6 | 17 | 2020-10-01 |
| 41 | Chrystel | Allwood | 1996-09-07 | 6958461692 | 41 | 3 | 25 | 2020-10-01 |
| 42 | Dyana | Clutterbuck | 1997-09-22 | 5842483886 | 42 | 1 | 1 | 2020-10-01 |
| 43 | Nikki | Edy | 1999-01-10 | 5096155315 | 43 | 6 | 25 | 2020-10-01 |
| 44 | Hendrik | Surr | 1997-04-05 | 2021255732 | 44 | 5 | 11 | 2020-10-01 |
| 45 | Marta | Bosch | 1998-09-28 | 4075136713 | 45 | 6 | 5 | 2020-10-01 |
| 46 | Garrik | Pell | 1999-04-14 | 3071057649 | 46 | 6 | 7 | 2020-10-01 |
| 47 | Stormi | Colbron | 1998-10-21 | 9968113654 | 47 | 3 | 28 | 2020-10-01 |
| 48 | Angelique | Iacivelli | 1995-06-07 | 9518365081 | 48 | 5 | 7 | 2020-10-01 |
| 49 | Zack | Hefforde | 1999-07-25 | 5455693035 | 49 | 1 | 29 | 2020-10-01 |
| 50 | Gusella | Pettiford | 1999-08-23 | 2425172721 | 50 | 4 | 3 | 2020-10-01 |

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Joey','Ironside','11/22/1995',1276234258,1,3,1);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Karlotta','Garraway','7/6/1997',2192431615,2,3,24);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Jerry','Stutte','12/18/1996',4125425783,3,1,17);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Yehudit','Rahill','1/15/1995',9939485406,4,2,29);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Cele','Crosetto','11/24/1998',3622733725,5,3,16);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Hazlett','Mowsdale','4/9/1995',1482883476,6,4,23);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Carlyn','Marks','12/27/1996',6129154080,7,5,20);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Ellis','Boatman','4/29/1997',8269707118,8,6,7);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Florina','Boyack','8/3/1997',9623352863,9,3,14);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Borg','Innett','9/3/1997',5256034960,10,1,19);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Sayres','Jennings','5/12/1996',8675076454,11,4,27);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Jarid','Sprull','11/2/1998',1391270091,12,2,6);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Elvera','Bannard','9/7/1996',7897232539,13,4,24);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Ody','Inggall','3/5/1995',6094734260,14,5,25);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Curcio','McWhan','7/29/1996',2394865847,15,6,11);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Connie','Sinnie','7/19/1995',1473936221,16,6,23);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Auroora','Nel','9/5/1996',2216400391,17,3,14);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Wendall','Rosendale','12/30/1999',1818120249,18,3,28);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Hadley','Bradbury','8/16/1996',6518067697,19,1,10);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Celine','Smales','7/11/1999',7106508130,20,2,10);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Jesselyn','Stevenson','5/16/1998',9231672206,21,2,22);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Corinna','Pinkney','1/16/1998',8323630067,22,5,29);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Orelle','Adamthwaite','7/26/1997',2539126766,23,3,17);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Howie','Seaman','12/1/1997',9888259627,24,2,4);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Sibyl','Corey','7/18/1996',4493239590,25,5,11);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Ruperta','Peaker','5/22/1999',5124781263,26,5,4);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Delmer','Roughey','4/21/1995',4175314364,27,3,22);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Gifford','O''Scannill','10/31/1996',3134783726,28,4,22);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Hedy','O''Hone','3/29/1998',7316228047,29,2,17);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Shalna','Hyde-Chambers','11/23/1999',7455116160,30,5,6);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Ferdie','Di Napoli','1/17/1995',1905908693,31,4,30);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Piper','Giacomuzzo','9/14/1998',5499340503,32,6,4);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Gerhardt','Schruurs','11/18/1999',8197494894,33,3,1);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Mellicent','Buncher','10/3/1996',4584525312,34,5,28);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Corette','Demead','9/17/1997',4909862137,35,5,17);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Jorgan','Barson','5/1/1997',6022309183,36,1,21);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Koral','Bowen','5/12/1998',4198817454,37,4,3);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Allissa','Kitter','8/17/1998',7328676920,38,5,7);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Townsend','Doughtery','4/13/1998',2639777958,39,4,7);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Yolane','Geratt','6/10/1998',2069585951,40,6,17);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Chrystel','Allwood','9/7/1996',6958461692,41,3,25);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Dyana','Clutterbuck','9/22/1997',5842483886,42,1,1);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Nikki','Edy','1/10/1999',5096155315,43,6,25);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Hendrik','Surr','4/5/1997',2021255732,44,5,11);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Marta','Bosch','9/28/1998',4075136713,45,6,5);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Garrik','Pell','4/14/1999',3071057649,46,6,7);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Stormi','Colbron','10/21/1998',9968113654,47,3,28);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Angelique','Iacivelli','6/7/1995',9518365081,48,5,7);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Zack','Hefforde','7/25/1999',5455693035,49,1,29);**

**INSERT INTO enrol.Student(StudentFirstName,StudentLastName,StudentDOB,StudentMobile,StudentRollNo,DepartmentID,AddressID) VALUES ('Gusella','Pettiford','8/23/1999',2425172721,50,4,3);**

1. Write the following Query based on the above datasets.
   1. List all the Student information from the Student table.

**select \* from enrol.student;**

* 1. List all the Department information from the Department table.

**select \* from enrol.Department;**

* 1. List all the Lecturer information from the Lecturer table.

**select \* from enrol.Lecturer;**

1. List all the Address information from the Address table.

**select \* from enrol.Address;**

1. List the StudentFullName, StudentDOB, StudentMobile from Student [StudentFullName=StudentFirstName + ‘ ‘ + StudentLastName]

**SELECT**

**CONCAT(enrol.Student.StudentFirstName,' ',enrol.Student.StudentLastName) AS StudentFullName,StudentDOB,**

**StudentMobile**

**FROM enrol.Student;**

1. List the StudentID, StudentFirstName, StudentLastName, StudentDOB, StudentMobile from Student StudentRollNo in AddressID 7.

**SELECT**

**StudentID, StudentFirstName, StudentLastName, StudentDOB, StudentMobile StudentDOB,StudentMobile**

**FROM**

**enrol.Student**

**WHERE AddressID=7**

**;**

1. List all the student information whose first name is start with 'B'

**SELECT**

**StudentID, StudentFirstName, StudentLastName, StudentDOB, StudentMobile StudentDOB,StudentMobile**

**FROM**

**enrol.Student**

**WHERE StudentFirstName Like 'B%'**

**;**

1. List all the student information whose first name is start and end with 'A']

**SELECT**

**StudentID, StudentFirstName, StudentLastName, StudentDOB, StudentMobile StudentDOB,StudentMobile**

**FROM**

**enrol.Student**

**WHERE StudentFirstName Like 'A%%A'**

1. Count the number of Student from Student table whose DepartmentID 6.

**SELECT**

**COUNT(\*) AS StudentInDepartment6**

**FROM**

**enrol.Student**

**WHERE DepartmentID=6**

;

1. List all the StudentFullName, StudentAge, StudentMobile from Student [StudentFullName= StudentFirstName + ‘ ‘ + StudentLastName]

[StudentAge= Current date – DOB (in Years)]

**SELECT**

**CONCAT(enrol.Student.StudentFirstName,' ',enrol.Student.StudentLastName) AS StudentFullName,**

**DATEDIFF(year, StudentDOB, GETDATE()) AS StudentAge,**

**StudentMobile**

**FROM enrol.Student;**

1. List all the StudentFullName, StudentAge, StudentMobile whose Age>23 from Student [StudentFullName= StudentFirstName + ‘ ‘ + StudentLastName]

[StudentAge= Current date – DOB (in Years)]]

**SELECT**

**CONCAT(enrol.Student.StudentFirstName,' ',enrol.Student.StudentLastName) AS StudentFullName,**

**DATEDIFF(year, StudentDOB, GETDATE()) AS StudentAge,**

**StudentMobile**

**FROM enrol.Student**

**WHERE**

**DATEDIFF(year, StudentDOB, GETDATE())>23;**

1. List all the StudentFullName, StudentAge, StudentMobile whose Age is either 21 or 23 from Student [StudentFullName= StudentFirstName + ‘ ‘ + StudentLastName]

[StudentAge= Current date – DOB (in Years)]

**SELECT**

**CONCAT(enrol.Student.StudentFirstName,' ',enrol.Student.StudentLastName) AS StudentFullName,**

**DATEDIFF(year, StudentDOB, GETDATE()) AS StudentAge,**

**StudentMobile**

**FROM enrol.Student**

**WHERE**

**DATEDIFF(year, StudentDOB, GETDATE())=21 OR DATEDIFF(year, StudentDOB, GETDATE())=23;**

1. List all the LecturerID, LecturerName, LecturerHighestQualification, LecturerAge from Lecturer.

**SELECT**

**LecturerID, LecturerName, LecturerHighestQualification, LecturerAge**

**FROM enrol.Lecturer;**

1. List all the LecturerID, LecturerName, LecturerHighestQualification, LecturerAge from Lecturer whose HighestQualification is either “MS” or “PhD”.

**SELECT**

**LecturerID, LecturerName, LecturerHighestQualification, LecturerAge**

**FROM enrol.Lecturer**

**WHERE LecturerHighestQualification='MS' or LecturerHighestQualification='PhD';**

1. List all the lecturer information who belongs to DepartmentID 2.

**SELECT**

**\***

**FROM enrol.Lecturer**

**WHERE DepartmentID=2;**

1. List all the lecturer information whose name end with “R”.

**SELECT**

**\***

**FROM enrol.Lecturer**

**WHERE LecturerName Like '%R';**

1. List all the lecturer information whose name either start or end with “E”.

SELECT

\*

FROM enrol.Lecturer

WHERE LecturerName Like '%E' OR LecturerName Like '%E';

1. List all the lecturer name is in capital letter.

**SELECT**

**UPPER(LecturerName) AS LecturerName**

**FROM enrol.Lecturer;**

1. Display 5 character from the lecturer name along with LecturerID and LecturerHighestQualification.

**SELECT**

**SUBSTRING(LecturerName,1,5) AS LecturerName,LecturerID,LecturerHighestQualification**

**FROM enrol.Lecturer;**

1. List LecturerID, LecturerName, LecturerHighestQualification, LecturerAge(in year) [LecturerAge= Current Date – LecturerAge)] (in year).

**SELECT**

**LecturerID, LecturerName, LecturerHighestQualification,LecturerAge,**

**DATEDIFF(year, LecturerAge, GETDATE()) AS LecturerAge**

**FROM enrol.Lecturer**

1. List DepartmentID, DepartmentName, DepartmentDescription, DepartmentCapacity from Department.

**SELECT DepartmentID, DepartmentName, DepartmentDescription, DepartmentCapacity from enrol.Department;**

1. List all the Department information who’s DepartmentName is “ECE”.

SELECT \* FROM enrol.Department where DepartmentName='ECE';

1. List all DepartmentName, DepartmentDescription, DepartmentCapacity from Department whose capacity is greater than 60.

SELECT DepartmentName, DepartmentDescription, DepartmentCapacity

FROM enrol.Department

where DepartmentCapacity>60;

1. List all AddressID, StreetAddress, City, State, PostalCode, Country from Address.

SELECT AddressID, StreetAddress, City, State, PostalCode, Country

FROM enrol.Address;

1. List all AddressID, StreetAddress, City, State, PostalCode, Country from Address who belongs to “Poland” country.

SELECT AddressID, StreetAddress, City, State, PostalCode, Country

FROM enrol.Address where Country='Poland';

1. List all the Address information whose state is null.

SELECT AddressID, StreetAddress, City, State, PostalCode, Country

FROM enrol.Address where state is NULL;

1. List all the Address information whose PostalCode is not null.

SELECT AddressID, StreetAddress, City, State, PostalCode, Country

FROM enrol.Address where PostalCode is NOT NULL;

1. List all the Address information whose City name is "Honda" and Country name is "Colombia"

SELECT AddressID, StreetAddress, City, State, PostalCode, Country

FROM enrol.Address where City='Honda' and Country='Colombia';

1. Write the following Query based on the above datasets.
   1. List unique DOB from Student.

SELECT DISTINCT (StudentDOB) FROM enrol.Student;

* 1. List unique DepartmentName from Department.

SELECT DISTINCT (DepartmentName ) FROM enrol.Department;

* 1. List unique Country name from Address.

SELECT DISTINCT (Country ) FROM enrol.Address;

* 1. List unique State name from Address.

SELECT DISTINCT (State ) FROM enrol.Address;

* 1. List unique City name from Address.

SELECT DISTINCT (City) FROM enrol.Address;

* 1. List all the LecturerID, LecturerName, LecturerHighestQualification, LecturerYearService from Lecturer [LecturerYearService= Current Date – LecturerAge] (in year).

SELECT

LecturerID, LecturerName, LecturerHighestQualification,LecturerAge,

DATEDIFF(year, LecturerAge, GETDATE()) AS LecturerYearService

FROM enrol.Lecturer

* 1. List all the LecturerID, LecturerName, LecturerHighestQualification, LecturerType from Lecturer [LecturerType= if LecturerYearService< 5 then "Begining Level Experience" else if LecturerYearService>= 5 and LecturerYearService<10 then "Mid Level experience" else "Experienced".

**SELECT LecturerID, LecturerName, LecturerHighestQualification,LecturerAge,**

**DATEDIFF(year, LecturerAge, GETDATE()) AS LecturerYearService,**

**CASE WHEN DATEDIFF(year, LecturerAge, GETDATE())  < 5 THEN 'Begining Level Experience'**

**WHEN  DATEDIFF(year, LecturerAge, GETDATE())>= 5 and DATEDIFF(year, LecturerAge, GETDATE())< 10 THEN 'Mid Level experience'**

**ELSE 'Experienced'**

**END AS LecturerType**

**FROM enrol.Lecturer;**

1. Write the following Query based on the above datasets.
   1. Display all Student and their Department Information based on the relationship.

SELECT S.StudentFirstName,S.StudentLastName,S.StudentDOB,S.StudentMobile,S.StudentRollNo,D.DepartmentName

FROM enrol.Student S, enrol.Department D

WHERE S.DepartmentID=D.DepartmentID;

* 1. Display all Student and their Address Information based on the relationship.

SELECT \*

FROM enrol.Student S, enrol.Address A

WHERE S.AddressID=A.AddressID;

* 1. Display all Department and their Lecturer Information based on the relationship.

SELECT \*

FROM enrol.Lecturer L, enrol.Department D

WHERE L.DepartmentID=D.DepartmentID;

* 1. Display all Student with their Department with Lecturer Information based on the relationship.

SELECT \*

FROM enrol.Lecturer L, enrol.Department D, enrol.Student S

WHERE S.DepartmentID=D.DepartmentID and D.LecturerID=L.LecturerID;

* 1. Display all Student with their Address and Department Information based on the relationship.

SELECT

\*

FROM enrol.Student S

JOIN enrol.Address A

ON S.AddressID = A.AddressID

JOIN enrol.Department D

ON S.DepartmentID

* 1. Display all Student with Address, Department and Lecturer Information based on the relationship.

SELECT

DISTINCT S.StudentFirstName,S.StudentLastName,S.StudentDOB,S.StudentMobile,S.StudentRollNo,

A.City,A.Country, A.State, A.StreetAddress, A.PostalCode,

D.DepartmentName, D.DepartmentName, D.DepartmentDescription,D.DepartmentCapacity,

L.LecturerName, L.LecturerHighestQualification, L.LecturerAge

FROM enrol.Student S

JOIN enrol.Address A

ON S.AddressID = A.AddressID

JOIN enrol.Department D

ON S.DepartmentID =D.DepartmentID

JOIN enrol.Lecturer L

ON D.DepartmentID =L.DepartmentID;

* 1. Display all Student with Address, Department and Lecturer Information who belongs to either “ME” or “ECE” department.

SELECT

DISTINCT S.StudentFirstName,S.StudentLastName,S.StudentDOB,S.StudentMobile,S.StudentRollNo,

A.City,A.Country, A.State, A.StreetAddress, A.PostalCode,

D.DepartmentName, D.DepartmentName, D.DepartmentDescription,D.DepartmentCapacity,

L.LecturerName, L.LecturerHighestQualification, L.LecturerAge

FROM enrol.Student S

JOIN enrol.Address A

ON S.AddressID = A.AddressID

JOIN enrol.Department D

ON S.DepartmentID =D.DepartmentID

JOIN enrol.Lecturer L

ON D.DepartmentID =L.DepartmentID

WHERE D.DepartmentName='ME' Or D.DepartmentName='ECE'

* 1. Display Student with Department and their Lecturer information based on the LecturerHighestQualification either “MS” or “PhD”.

SELECT

DISTINCT S.StudentFirstName,S.StudentLastName,S.StudentDOB,S.StudentMobile,S.StudentRollNo,

A.City,A.Country, A.State, A.StreetAddress, A.PostalCode,

D.DepartmentName, D.DepartmentName, D.DepartmentDescription,D.DepartmentCapacity,

L.LecturerName, L.LecturerHighestQualification, L.LecturerAge

FROM enrol.Student S

JOIN enrol.Address A

ON S.AddressID = A.AddressID

JOIN enrol.Department D

ON S.DepartmentID =D.DepartmentID

JOIN enrol.Lecturer L

ON D.DepartmentID =L.DepartmentID

WHERE L.LecturerHighestQualification='MS' Or L.LecturerHighestQualification='PhD'

* 1. Display Student with Department and Address Information, where student belongs to “Thailand” country.

SELECT

DISTINCT S.StudentFirstName,S.StudentLastName,S.StudentDOB,S.StudentMobile,S.StudentRollNo,

A.City,A.Country, A.State, A.StreetAddress, A.PostalCode,

D.DepartmentName, D.DepartmentName, D.DepartmentDescription,D.DepartmentCapacity,

L.LecturerName, L.LecturerHighestQualification, L.LecturerAge

FROM enrol.Student S

JOIN enrol.Address A

ON S.AddressID = A.AddressID

JOIN enrol.Department D

ON S.DepartmentID =D.DepartmentID

JOIN enrol.Lecturer L

ON D.DepartmentID =L.DepartmentID

WHERE A.Country='Thailand' ;

* 1. Display Count of Student, Department wise.

SELECT D.DepartmentName,Count(\*) AS Count

FROM enrol.Student S

JOIN enrol.Department D

ON S.DepartmentID =D.DepartmentID

Group BY D.DepartmentName

* 1. Display Count of Lecturer, Department wise.

SELECT D.DepartmentName,Count(\*) AS LecturerCount

FROM enrol.Lecturer L

JOIN enrol.Department D

ON L.DepartmentID =D.DepartmentID

Group BY D.DepartmentName

* 1. Display Count of Student, Country wise.

SELECT A.Country,Count(\*) AS StudentCount

FROM enrol.Student S

JOIN enrol.Address A

ON S.AddressID =A.AddressID

Group BY A.Country

1. Write the following Query based on the above datasets.
   1. Create new table StudCopy and copy all records from Student table.

SELECT \* INTO enrol.StudCopy

FROM

enrol.Student;

* 1. Create a new table DeptCopy and copy only the schema from Department table.

SELECT \* INTO enrol.DeptCopy

FROM

enrol.Department where 1=2;

* 1. Create a new table DepartmentCopy and copy all records from Department table.

SELECT \* INTO enrol.DepartmentCopy

FROM

enrol.Department;

* 1. Create a new table AddrCopy and copy only the schema from Address table.

SELECT \* INTO enrol.AddrCopy

FROM

enrol.Address

Where 1=2;

* 1. Create a new table AddrCopy and copy all the records from Address table.

SELECT \* INTO enrol.AddrCopy

FROM

enrol.Address;

* 1. Create a new table LecturerCopy and copy all the records from Lecturer table.

SELECT \* INTO enrol.LecturerCopy

FROM

enrol.Lecturer;

1. Write the following Query based on the above datasets.
   1. Delete all the records from LecturerCopy table.

DELETE FROM enrol.LecturerCopy;

* 1. Delete all the student information for the students who belong to “IT” department.

DELETE S FROM enrol.Student S

JOIN enrol.Department D

ON S.DepartmentID = D.DepartmentID

WHERE D.DepartmentName='IT';

* 1. Delete all the student information for the students who belong to “Indonesia” country.

DELETE S FROM enrol.Student S

JOIN enrol.Address A

ON S.AddressID = A.AddressID

WHERE A.Country='Indonesia';

* 1. Delete all the student information for the student who belongs to “Nanshi” city.

DELETE S FROM enrol.Student S

JOIN enrol.Address A

ON S.AddressID = A.AddressID

WHERE A.City='Nanshi';

* 1. Delete all the student information for the student who belongs to “Bretagne” state.

DELETE S FROM enrol.Student S

JOIN enrol.Address A

ON S.AddressID = A.AddressID

WHERE A.State='Bretagne';

1. Write the following Query based on the above datasets.
   1. Update StudentMobile for those students who belongs to Department “ME”.

UPDATE S SET S.StudentMobile ='123456789'

FROM enrol.Student S, enrol.Department D

WHERE

S.DepartmentID=D.DepartmentID and

D.DepartmentName='ME';

* 1. Update Student DepartmentID as 3, for the StudentID=42.

UPDATE S SET S.DepartmentID=3

FROM enrol.Student S

WHERE S.StudentID=42;

* 1. Update LecturerHighestQualification as “PHd” for the Lecturer whose LecturerHighestQualification= “PhD”.

UPDATE L SET L.LecturerHighestQualification='PhD'

FROM enrol.Lecturer L

WHERE L.LecturerHighestQualification='PHd';

* 1. Update PostalCode as “00000” for the Address which contain NULL as a PostalCode.

UPDATE A SET A.PostalCode='00000'

FROM enrol.Address A

WHERE A.PostalCode is NULL;

* 1. Update StudentLastName as “Paul” for the Student whose Name is “Jerry”.

UPDATE S SET S.StudentLastName='Paul'

FROM enrol.Student S

WHERE S.StudentFirstName='Jerry';