Chapter 1

**Name : Kinz UL Iman**

**ID :BSCS-2023-076**

**Instructor : SIR,Hibatullah**

**Note (email subject) : <id> - CS3A – DBMS – LAB 01 -**

**Email : Hibatullah902@gmail.com**

**Deadline : 18/02/2024**

**Week 1: Getting started with MySQL**

**Learning Objectives:**

The objectives of this first experiment are:

1. Getting familiarized with the Workbench IDE.
2. Getting familiarized with creation of database
3. Getting familiarized with creation of tables
4. Getting familiarized with storing of values in tables
5. Performing column manipulation operations

**Exercise 1:**

Create a Database named GCUH and verify its presence.

Answer:

Code:

|  |
| --- |
| create database gcuh  use gcuh |

Output(Screenshot):

|  |
| --- |
|  |

**Exercise 2:**

Create three tables inside the GCUH database with five attributes in each table

Answer:

Code:

|  |
| --- |
| create database gcuh;  use gcuh;  create table total\_dep(  first\_dep\_name varchar(20),  sec\_dep\_name varchar(20),  third\_dep\_name varchar(20),  fouth\_dep\_name varchar(20),  fifth\_dep\_name varchar(20));  CREATE TABLE courses (  course\_id INT ,  course\_name VARCHAR(25),  department VARCHAR(50),  credits INT,  instructor\_name VARCHAR(255)  );  CREATE TABLE students (  student\_id INT ,  student\_name VARCHAR(255) ,  program VARCHAR(50),  semester INT,  email VARCHAR(100)  ) |

Output:

|  |
| --- |
|  |

**Exercise 3:**

Insert 5 values in each of the tables created earlier.

Code:

|  |
| --- |
| create database gcuh;  use gcuh;  create table total\_dep(  first\_dep\_name varchar(20),  sec\_dep\_name varchar(20),  third\_dep\_name varchar(20),  fouth\_dep\_name varchar(20),  fifth\_dep\_name varchar(20));  insert into total\_dep  values  ("cs","english","it","ds","ai"),  ("cs","english","it","ds","ai"),  ("cs","english","it","ds","ai"),  ("cs","english","it","ds","ai"),  ("cs","english","it","ds","ai");  CREATE TABLE courses (  course\_id INT ,  course\_name VARCHAR(25),  department VARCHAR(50),  credits INT,  instructor\_name VARCHAR(255)  );  insert into courses  values  (101,"database","cs",4,"sir,habatullah"),  (101,"database","cs",4,"sir,habatullah"),  (101,"database","cs",4,"sir,habatullah"),  (101,"database","cs",4,"sir,habatullah"),  (101,"database","cs",4,"sir,habatullah");  CREATE TABLE students (  student\_id INT ,  student\_name VARCHAR(255) ,  program VARCHAR(50),  semester INT,  email VARCHAR(100)  );  insert into students  values  (076,"kinza","mysql",3,"ajhajksh@gmail.com"),  (076,"kinza","mysql",3,"ajhajksh@gmail.com"),  (076,"kinza","mysql",3,"ajhajksh@gmail.com"),  (076,"kinza","mysql",3,"ajhajksh@gmail.com"),  (076,"kinza","mysql",3,"ajhajksh@gmail.com") |

Output:

|  |
| --- |
|  |

**Exercise 4:**

Rename any of the column in the table created above.

For example change name column to first name

Answer:

Code:

|  |
| --- |
| alter table students  rename column student\_id to roll\_num |

Output:

|  |
| --- |
|  |

**Exercise 5:**

Add new column to earlier created table in the at the second location

Answer:

Code:

|  |
| --- |
| ALTER TABLE total\_dep  ADD column NEW\_COLUMN INT  AFTER first\_dep\_name |

Output:

|  |
| --- |
|  |

**Name : kinz ul iman**

**ID :BSCS-2023-076**

**Instructor :SIR, HIBATULLAH**

**Note (email subject) : <id> - CS3A – DBMS – LAB 02 -**

**Email : Hibatullah902@gmail.com**

**Deadline : 26/02/2024**

**Week 2: Operations on Records**

**Learning Objectives:**

The objectives of this first experiment are:

1. Getting familiarized with the different implementations of select.
2. Getting familiarized with the update table queries.
3. Getting familiarized with the delete data from table queries.
4. Getting familiarized with the commit and rollback queries

**Exercise 1:**

Create a Database named GCUH, create table students and subjects and view only subject title.

Answer:

Code:

|  |
| --- |
| create table students(  student\_name varchar(20),  subjects varchar (20)  )  insert into students  values  ("ALI","math"),  ("kinza","computer"),  ("tooba","biotech")  select subjects from students |

Output(Screenshot):

|  |
| --- |
|  |

**Exercise 2:**

From the previous table, add column cgpa if its not already present and view students whose cgpa is above 2.0

Answer:

Code:

|  |
| --- |
| alter table students  add column CGPA decimal(3,2)  update students  set cgpa=3.87  where subjects ="computer"  update students  set cgpa=2.00  where subjects ="maths"  update students  set cgpa=2.00  where subjects ="maths"  update students  set cgpa=1.27  where subjects ="biotech"  select cgpa from students  where cgpa>2.00; |

Output:

|  |
| --- |
|  |

**Exercise 3:**

Update the name and cgpa of students.

Code:

|  |
| --- |
| update students  set student\_name="ali"  where subjects="math"  update students  set cgpa=2.0  where subjects="math" |

Output:

|  |
| --- |
|  |

**Exercise 4:**

Delete one of the record from the table.

Answer:

Code:

|  |
| --- |
| delete from students  where subjects="computer" |

Output:

|  |
| --- |
|  |

**Exercise 5:**

Implement commit queries and show changes before and after rollback.

Answer:

Code:

|  |
| --- |
| Commit;  Rollback; |

Output:

|  |
| --- |
|  |

**Name : kinzuliman**

**ID :bscs-2023-076**

**Instructor :sir,hibatullah**

**Note (email subject) : <id> - CS3A – DBMS – LAB 03 -**

**Email : Hibatullah902@gmail.com**

**Deadline : 29/02/2024**

**Week 3: Constraints**

**Learning Objectives:**

The objectives of this first experiment are:

1. Getting familiarized with the check
2. Getting familiarized with the not null.

**Exercise 1:**

Create a table employee (with attributes: name, age, salary).   
put constraint for salary to be greater than 0.

Answer:

Code:

|  |
| --- |
| use gcuh;  create table employee(  name varchar (20),  age int ,  salary varchar(20)  check (salary>=0));  insert into employee  values  ("sara",20,-8),  ("saira",20,20000),  ("sana",20,20000)  ////////////////////////  Without error  ////////////////////////  create database gcuh2;  use gcuh;  create table employee(  name varchar (20),  age int ,  salary varchar(20)  check (salary>=0));  insert into employee  values  ("sara",20,12000),  ("saira",20,20000),  ("sana",20,20000) |

Output(Screenshot):

|  |
| --- |
|  |

**Exercise 2:**

Create a tablemarksheet (with attributes: roll\_no, std\_name, marks ) and put constraint on marks to lie between 0 and 100.

Answer:

Code:

|  |
| --- |
| create table marksheet(  roll\_no int,  std\_name varchar(20),  marks int,  check (marks between '0'and '100' ));  insert into marksheet  values  (076,"kinza",100),  (010,"urwa",88) |

Output:

|  |
| --- |
|  |

**Exercise 3:**

Create a table bank\_customers (with attributes: c\_name, age, balance). Put constraint on name to be less than 20 digits and name not being left empty at the time of filling the table

Code:

|  |
| --- |
| CREATE TABLE bank\_customers (  c\_name VARCHAR(20) NOT NULL,  age INT,  balance INT,  CHECK (LENGTH(c\_name) < 20)  );  INSERT INTO bank\_customers  VALUES(  "KINZULIMANMHMMADALI",20,4200) |

Output:

|  |
| --- |
|  |

**Exercise 4:**

Create a table customers\_list (with attributes: c\_name, phone\_no). The phone number must be exactly 12 digits long. For example: 923170001234

Answer:

Code:

|  |
| --- |
| Create database uni;  use uni;  create table customers\_list  ( c\_name varchar(10),  phone\_no char(12) primary key  );  insert into customers\_list  values  ("kinza",'012345678911') |

Output:

|  |
| --- |
|  |

**Name : kinz ul iman**

**ID :bscs-20203-076**

**Instructor : sir,hibatullah**

**Note (email subject) : <id> - CS3 – DBMS – LAB 04 -**

**Email : Hibatullah902@gmail.com**

**Deadline :**

**Week 4: Constraints II & DB Security**

**Learning Objectives:**

The objectives of these experiment are:

1. Getting familiarized with unique constraint
2. Getting familiarized with auto increment constraint.
3. Getting familiarized with encryption

**Exercise 1:**

Create a table students (with attributes: roll\_no, name, cgpa, contact).   
put constraint for all the roll\_no to be unique

Answer:

Code:

|  |
| --- |
| create database university;  use university;  create table students(  roll\_no int unique,  name varchar(10),  cgpa decimal(3,2),  contact int  );  insert into students  values  (11,"kinza",8.5,0123456789),  (11,"kinza",8.5,0123456789); |

Output(Screenshot):

|  |
| --- |
|  |

**Exercise 2:**

Create a tablemarksheet (with attributes: ans\_sheet\_no, roll\_no, std\_name, marks ) and put constraint for ans\_sheet\_no & roll\_no to be unique in a single constraint condition.

Answer:

Code:

|  |
| --- |
| create database university;  use university;  create table students(  ans\_sheet\_no int,  roll\_no int,  std\_name varchar(10),  marks int ,  unique(ans\_sheet\_no,roll\_no)  );  insert into students  values  (11,076,"kinza",92),  (11,076,"kinza",92) |

Output:

|  |
| --- |
|  |

**Exercise 3:**

Create a table bank\_customers (with attributes: c\_id c\_name, age, balance). The database should automatically increment the c\_id after 1st input of record.

Code:

|  |
| --- |
| create database university;  use university;  create table bank\_customers(  c\_id int auto\_increment primary key,  c\_name varchar(10),  age int,  balance int  );  insert into bank\_customers  values  (1,"kinza",21,4500) |

Output:

|  |
| --- |
|  |

**Exercise 4:**

Create a table users(with attributes: email, password).

The email address must be unique

Password should be encrypted before saving in table.

Answer:

Code:

|  |
| --- |
| create database quik;  use quik;  Create table users  ( email varchar(220) unique ,  password varchar(256)  );  insert into users values  ("kinzashykh@gmail", SHA2('12345',256)  ) |

Output:

|  |
| --- |
|  |

**DBMS Week 6 LAB Tasks:**

Q. Create a table named customers with attributes customer\_id, c\_username ,customer\_name, contact\_no, balance. The id should be primary key and the username should be unique.

Tasks:

1. Show all the constraints applied on the table.

Code:

create database kinza;

use kinza;

create table customer

(

customer\_id int primary key,

c\_username varchar(100) unique,

customer\_name varchar(100),

contact\_no int,

balance int

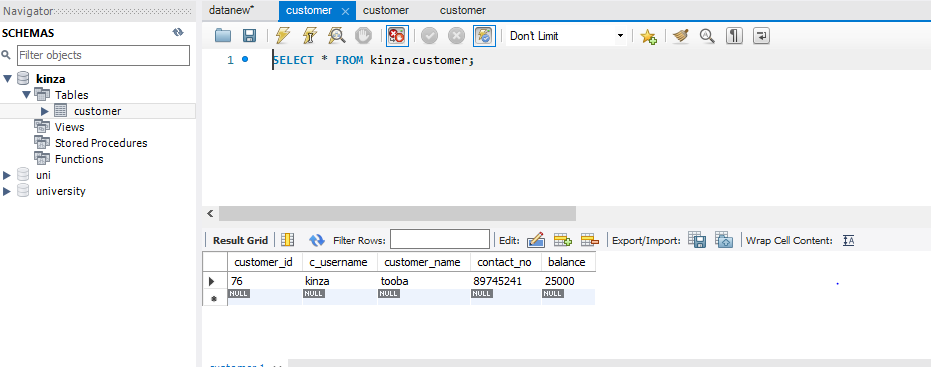
);

insert into customer

values

(076,"kinza","tooba",89745241,25000)

Output:



1. Remove primary key   
   Code:

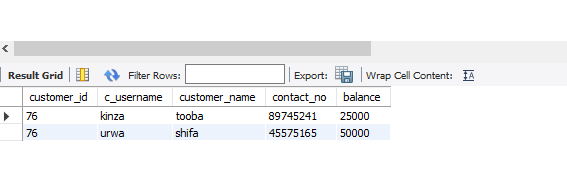
ALTER TABLE customer;

drop PRIMARY KEY;

insert into customer

values

(076,"urwa","shifa",45575165,50000)  
Output:

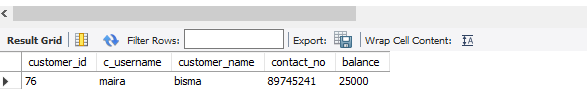


3)Search for customers whose name start with “b”.  
Code:

select \* FROM customer

WHERE customer\_name LIKE 'B%'

Output:

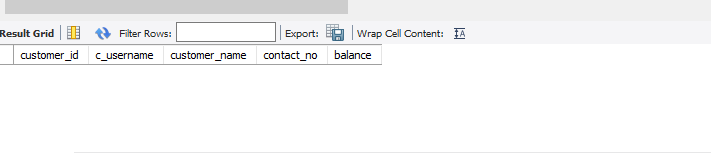


4)Search for customers, within their name should be pattern “li”.  
Code:

select \* FROM customer

where customer\_name like 'li%'

Output:



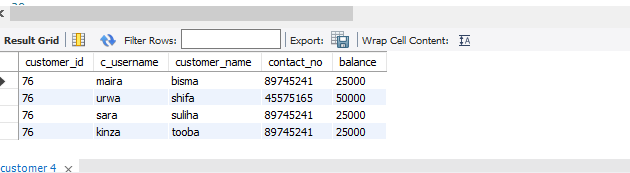
5)Show customers in ascending order of their names.

Code:

select\*from customer

order by customer\_name asc

Output:

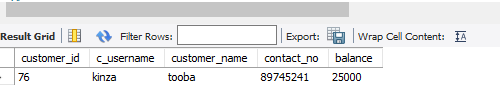


6)Show customers whose account has lowest balance.

Code:

SELECT \*FROM customer

ORDER BY balance ASC

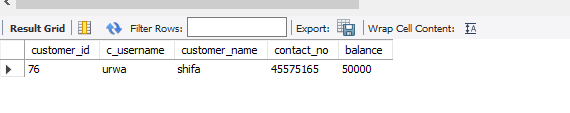
LIMIT 1  
Output:  


7)Show customers whose account has highest balance.

Code:

SELECT \*FROM customer

ORDER BY balance DESC

LIMIT 1  
Output:  


8)Show average balance of customers.  
Code:

select avg(balance)from customer  
Output:



**TASK : 7**

**Name : kinz ul iman**

**ID : BSCS-2023-076**

**Instructor : SIR HIBATULLAH**

**Note (email subject) : <id> - CS3A – DBMS – LAB 01 -**

**Email : Hibatullah902@gmail.com**

**Deadline : 22/05/2024**

**Week 7: Groups**

**Exercise :1**

**Q. Write a query to list the number of jobs available in the employees table.**

**ANSWER:**

**Code:**

|  |
| --- |
| use gc;  create table Office\_data(  First\_name varchar(30),  Last\_name varchar(40),  Phone\_no int,  job\_id int,  job\_title varchar(50),  Salary int  );  insert into Office\_data values  ("Umme shifa","shaikh",72345678,68,"Data scientist",450000),  ("kinza","shaikh",15678124,76,"Graphic Designer",78000),  ("Tooba","siddiqui",478123546,42,"Web Developer",20000),  ("urwa","Malik",96478329,10,"App Developer",50000),  ("Hasan","Bhabha",86789302,30,"Digital Marketer",67000),  ("Salal","Ghazi",86789302,32,"Big data engineer",1000000);  select job\_id,count(First\_name) as Job\_Available  from office\_data group by job\_id; |

**Output:**

|  |
| --- |
| **AFTER GROUPING:** |

**Exercise : 2**

**Q. Write a query to get the Total Salaries payable to employees.**

**ANSWER:**

**Code:**

|  |
| --- |
| select job\_id , sum(Salary)  from Office\_data group by job\_id; |

**Output:**

|  |
| --- |
|  |

**Exercise : 3**

**Q. Write a query to get the Minimum Salary from employee\_id table.**

**ANSWER:**

**Code:**

|  |
| --- |
| select job\_id , min(Salary)  from Office\_data group by job\_id; |

**Output:**

|  |
| --- |
|  |

**TASK : 8**

**Name : kinz ul iman**

**ID :BSCS-2023-076**

**Instructor : SIR,Hibatullah**

**Note (email subject) : <id> - CS3A – DBMS – LAB 01 -**

**Email : Hibatullah902@gmail.com**

**Deadline : 22/05/2024**

**Week 8: Foreign Key And Joins**

**Exercise :1**

**Create a table named "students" with a primary key column named "student\_id"**

**ANSWER:**

**Code:**

|  |
| --- |
| create database UNIVERSITY;  use UNIVERSITY;  create table students(  STUDENT\_ID int primary key,  STUDENT\_NAME varchar(30),  SIR\_NAME varchar(40),  COURSE\_ID int  ); |

**Output:**

|  |
| --- |
|  |

**Exercise :2**

**Q. Create a table named "Courses" with a primary key column named "Course\_id"**

**ANSWER:**

**Code:**

|  |
| --- |
| use university;  create table courses(  Course\_Id int primary key,  Course\_Name varchar(30),  Course\_Guider varchar(40)  ); |

**Output:**

|  |
| --- |
|  |

**Exercise :3**

**Q. Add a foreign key constraint to the "students" table referencing the "Course\_id" column in the "courses" table**

**ANSWER:**

**Code:**

|  |
| --- |
| create table students(  STUDENT\_ID int primary key,  STUDENT\_NAME varchar(60),  SIR\_NAME varchar(30),  COURSE\_ID int,  constraint fk\_id foreign key (Course\_id) references courses (Course\_id)  ); |

**Output:**

|  |
| --- |
|  |

**Exercise :4**

**Q. Insert data into the "students" and "courses" tables to establish relationships between the two tables**

**ANSWER:**

**Code:**

|  |
| --- |
| **INSERTING VALUES INTO STUDENTS TABLE:**  insert into students (STUDENT\_ID,STUDENT\_NAME,SIR\_NAME,COURSE\_ID)  values  (68,"Umme shifa ","shaikh",1),  (76,"kinza","shaikh",2),  (10,"urwa","malik",3),  (42,"Tooba","siddique",4),  (30,"Hasan","Bhabha",4),  (32,"Salal Ghazi","Qureshi",1);  **INSERTING VALUES INTO COURSES TABLE:**  insert into courses values  (1,"Data Structure","Sir Sohail Shah"),  (2,"Digital logic Design","Sir Allah Bachayo"),  (3,"Data Base Management System","Sir Hibatullah"),  (4,"Networking","Sir Ifran Baloach "),  (5,"Math","sir Sakhawat"); |

**Output:**

|  |
| --- |
| **STUDENTS TABLE:**    **COURSES TABLE:** |

**Exercise :5**

**Q. Perform a query to retrieve information from both tables using a JOIN operation to display data related to students and their courses**

**ANSWER:**

**Code:**

|  |
| --- |
| select \*from students join courses on students.Course\_id = courses.Course\_id |

**Output:**

|  |
| --- |
|  |

