CET2002B Database Management Systems

School of Computer Engineering and Technology

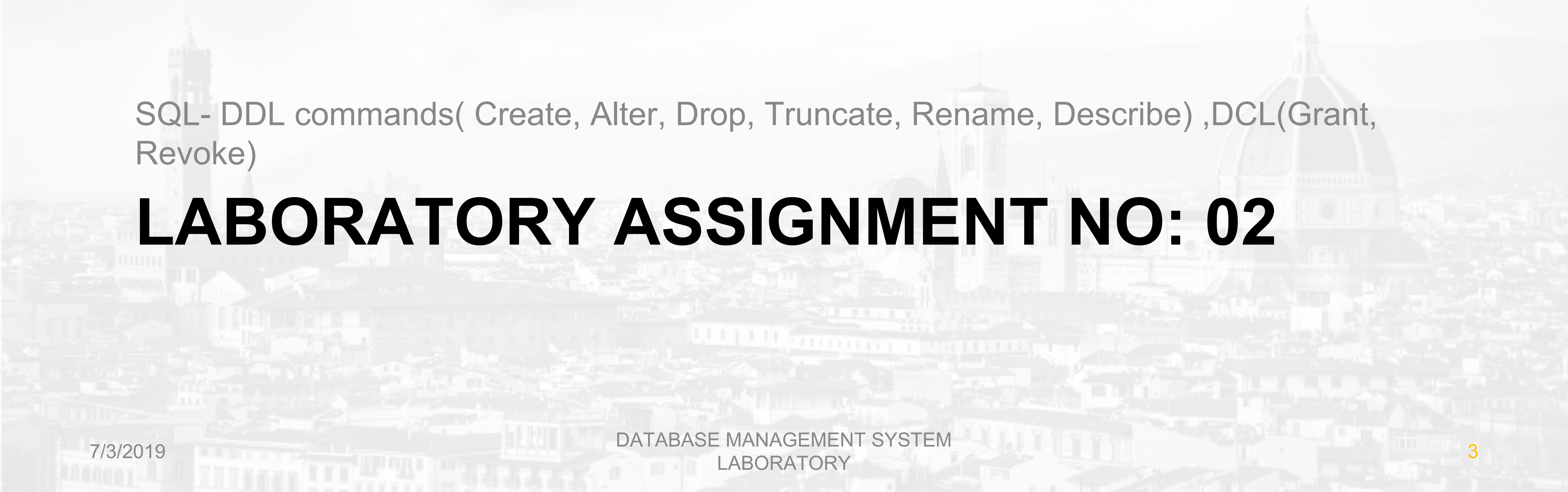
CET2002B Database Management Systems

- **Course Objectives:**

1) Understand and successfully apply logical database design principles, including E-R diagrams and database normalization.
2) Learn Database Programming languages and apply in DBMS application
3) Understand transaction processing and concurrency control in DBMS
4) Learn database architectures, DBMS advancements and its usage in advance application

- **Course Outcomes:**

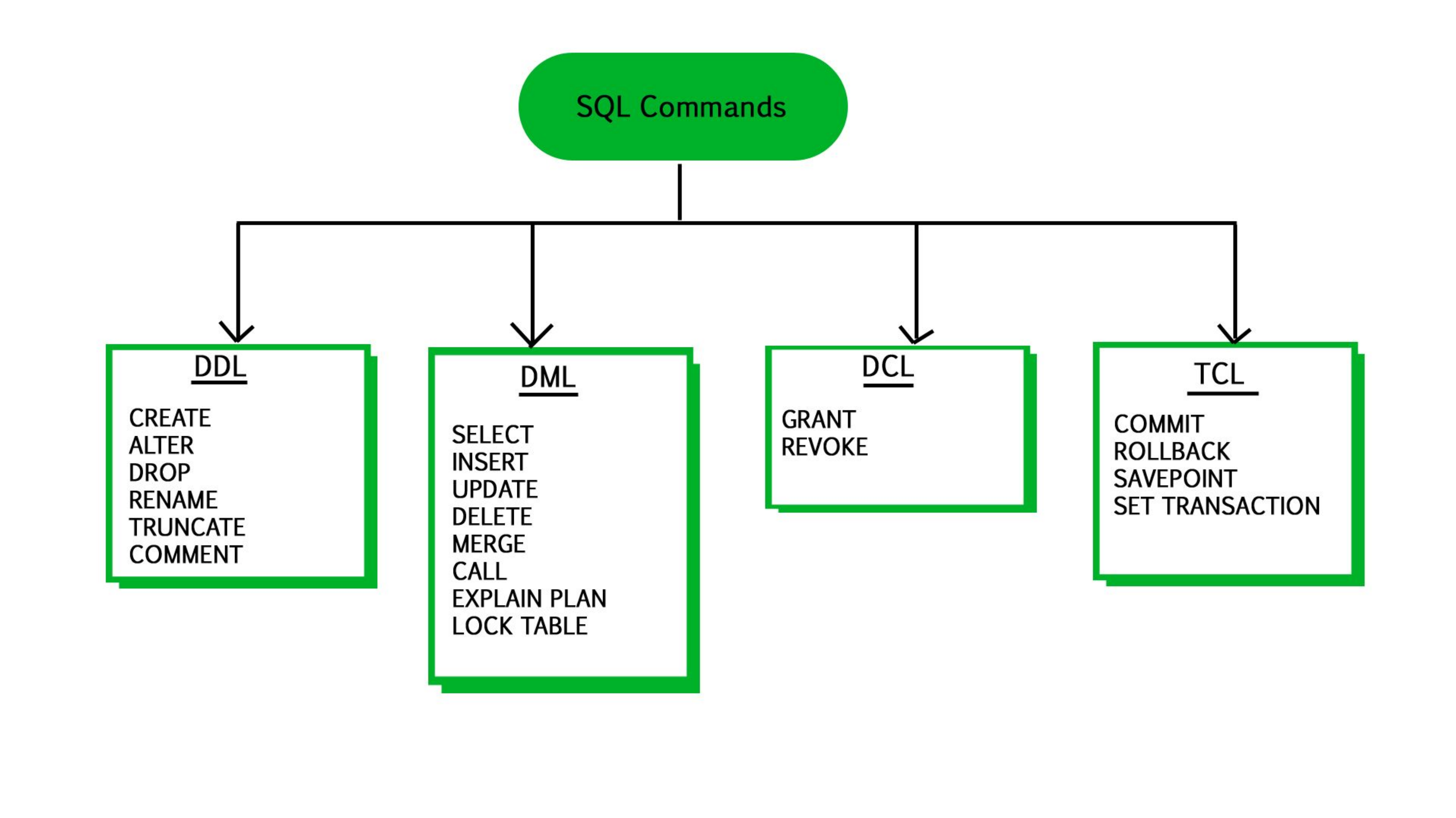
1) Design ER-models to represent simple database application scenarios and Improve the database design by normalization.
2) Design Database Relational Model and apply SQL , PLSQL concepts for database programming
3) Describe Transaction Processing and Concurrency Control techniques for databases
4) Identify appropriate database architecture for the real world database application



SQL- DDL commands(Create, Alter, Drop, Truncate, Rename, Describe) ,DCL(Grant, Revoke)

LABORATORY ASSIGNMENT NO: 02

SQL Statements Categories



SQL Commands

DDL

CREATE
ALTER
DROP
RENAME
TRUNCATE
COMMENT

DML

SELECT
INSERT
UPDATE
DELETE
MERGE
CALL
EXPLAIN PLAN
LOCK TABLE

DCL

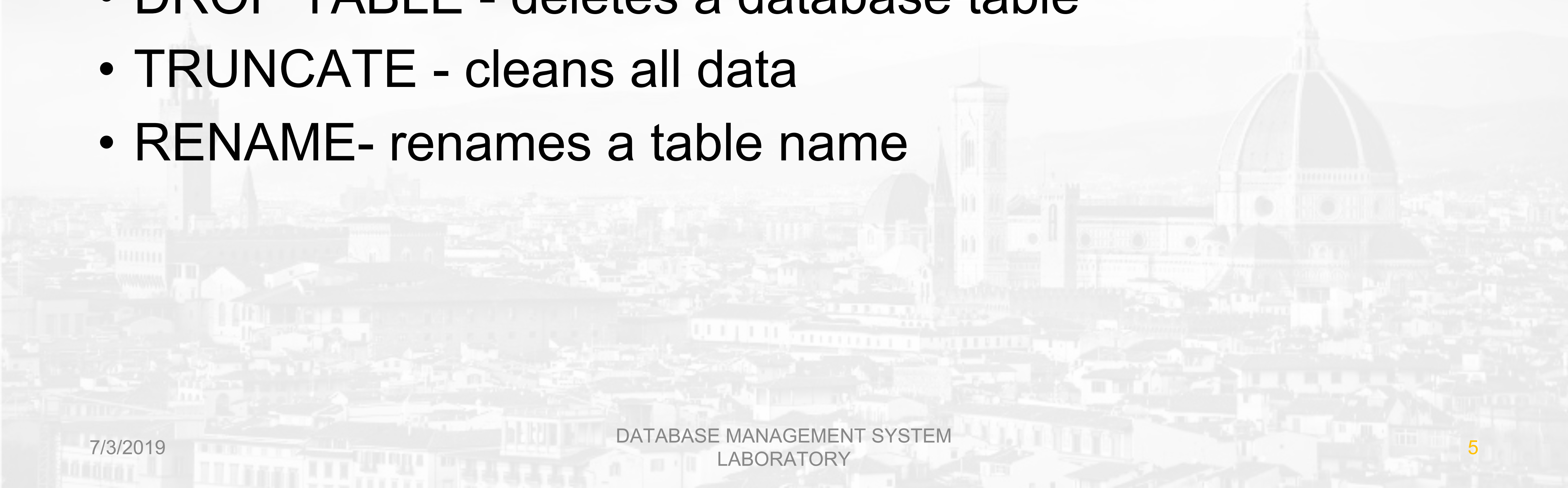
GRANT
REVOKE

TCL

COMMIT
ROLLBACK
SAVEPOINT
SET TRANSACTION

DDL Statements

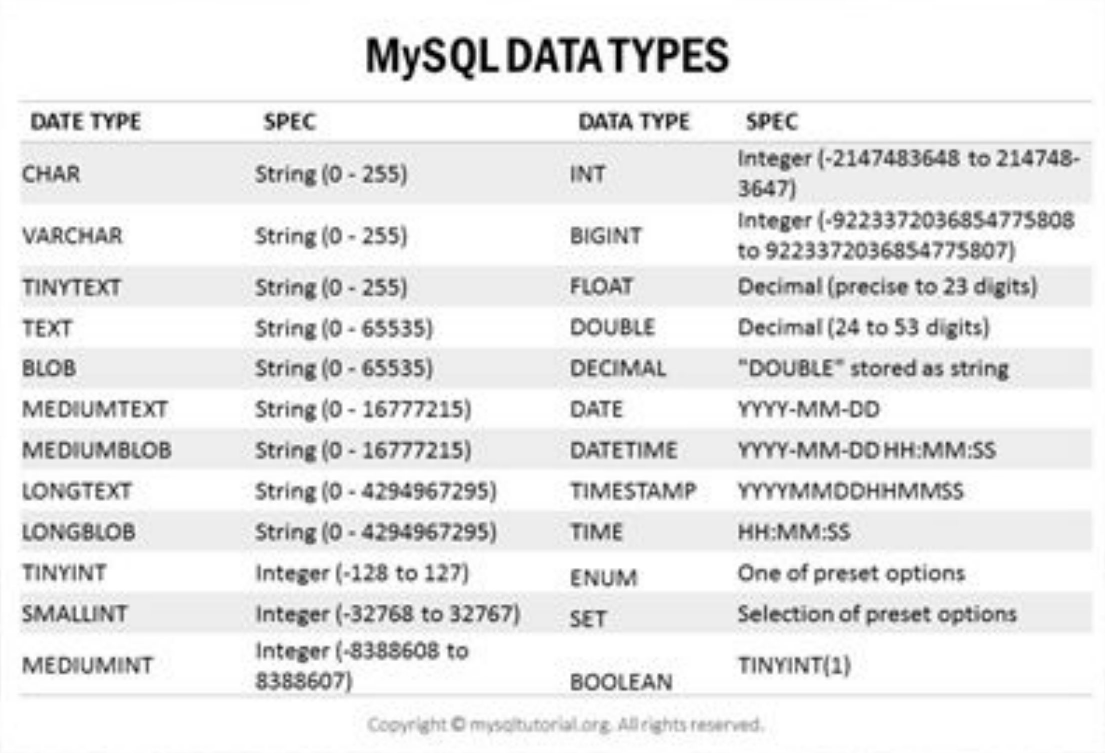
- CREATE TABLE - creates a new database table
- ALTER TABLE - alters (changes) a database table
- DROP TABLE - deletes a database table
- TRUNCATE - cleans all data
- RENAME- renames a table name



DCL Commands

- GRANT- allow specified users to perform specified tasks.
- REVOKE- cancel previously granted or denied permissions.

Datatypes



MySQL DATA TYPES

DATE TYPE	SPEC	DATA TYPE	SPEC
CHAR	String (0 - 255)	INT	Integer (-2147483648 to 2147483647)
VARCHAR	String (0 - 255)	BIGINT	Integer (-9223372036854775808 to 9223372036854775807)
TINYTEXT	String (0 - 255)	FLOAT	Decimal (precise to 23 digits)
TEXT	String (0 - 65535)	DOUBLE	Decimal (24 to 53 digits)
BLOB	String (0 - 65535)	DECIMAL	"DOUBLE" stored as string
MEDIUMTEXT	String (0 - 16777215)	DATE	YYYY-MM-DD
MEDIUMBLOB	String (0 - 16777215)	DATETIME	YYYY-MM-DD HH:MM:SS
LONGTEXT	String (0 - 4294967295)	TIMESTAMP	YYYYMMDDHHMMSS
LONGBLOB	String (0 - 4294967295)	TIME	HH:MM:SS
TINYINT	Integer (-128 to 127)	ENUM	One of preset options
SMALLINT	Integer (-32768 to 32767)	SET	Selection of preset options
MEDIUMINT	Integer (-8388608 to 8388607)	BOOLEAN	TINYINT(1)

DDL Commands

Syntax:
CREATE DATABASE database_name

DDL Creating a Table

- **Syntax**

```
CREATE TABLE table_name
(Column_name     datatype[(size)],
 Column_name     datatype[(size)],
)
```

- **Example**

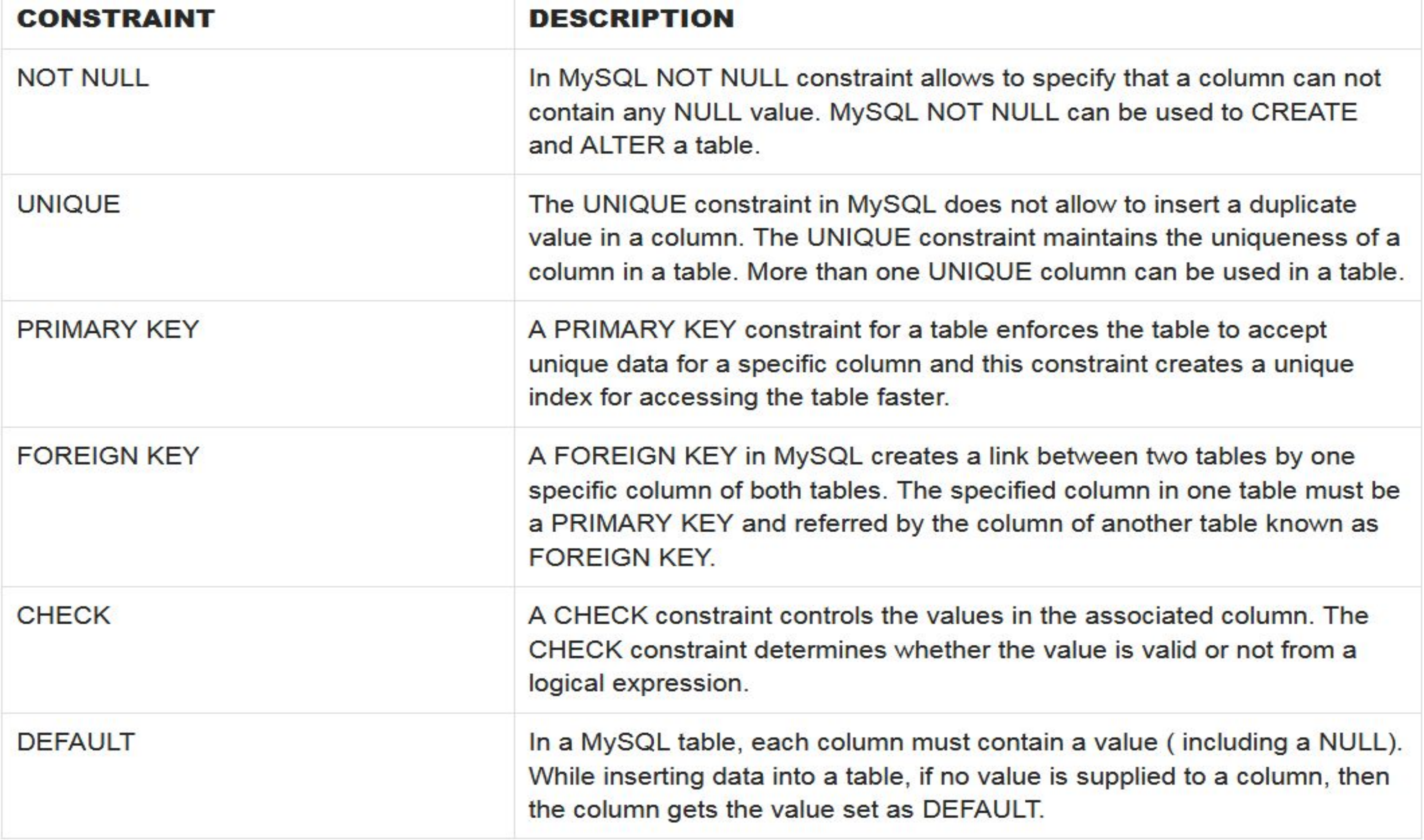
```
CREATE TABLE books
(ISBN     char(20),
Title        char(50),
AuthorID     Integer,
Price    float)
```

- Creates a table with four columns

MySQL CONSTRAINT

- MySQL CONSTRAINT is used to define rules to allow or restrict what values can be stored in columns. The purpose of inducing constraints is to enforce the integrity of a database.
- MySQL CONSTRAINTS are used to limit the type of data that can be inserted into a table.
- MySQL CONSTRAINTS can be classified into two types - column level and table level.
- The column level constraints can apply only to one column where as table level constraints are applied to the entire table.
- MySQL CONSTRAINT is declared at the time of creating a table.
- **MySQL CONSTRAINTs are :**
  - NOT NULL
  - UNIQUE
  - PRIMARY KEY
  - FOREIGN KEY
  - CHECK
  - DEFAULT

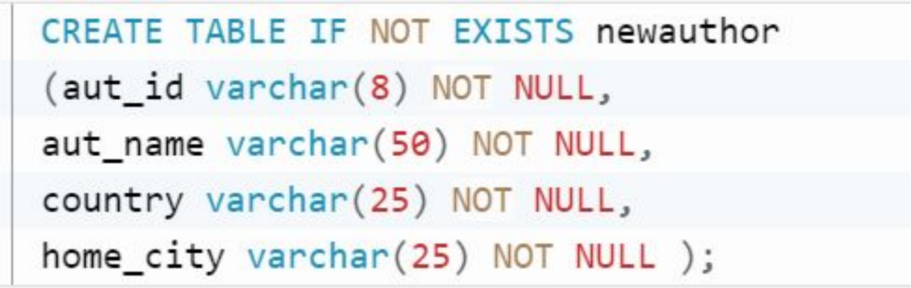
Constraints



CONSTRAINT	DESCRIPTION
NOT NULL	In MySQL NOT NULL constraint allows to specify that a column can not contain any NULL value. MySQL NOT NULL can be used to CREATE and ALTER a table.
UNIQUE	The UNIQUE constraint in MySQL does not allow to insert a duplicate value in a column. The UNIQUE constraint maintains the uniqueness of a column in a table. More than one UNIQUE column can be used in a table.
PRIMARY KEY	A PRIMARY KEY constraint for a table enforces the table to accept unique data for a specific column and this constraint creates a unique index for accessing the table faster.
FOREIGN KEY	A FOREIGN KEY in MySQL creates a link between two tables by one specific column of both tables. The specified column in one table must be a PRIMARY KEY and referred by the column of another table known as FOREIGN KEY.
CHECK	A CHECK constraint controls the values in the associated column. The CHECK constraint determines whether the value is valid or not from a logical expression.
DEFAULT	In a MySQL table, each column must contain a value (including a NULL). While inserting data into a table, if no value is supplied to a column, then the column gets the value set as DEFAULT.

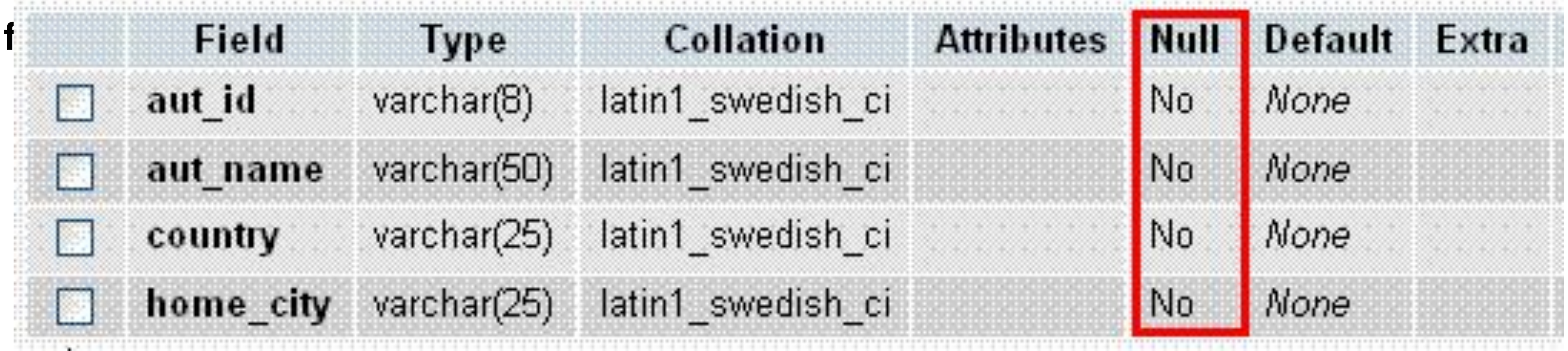
MySQL CREATE TABLE with NULL CONSTRAINT

- **Example**
  - If we want to create a table 'newauthor' where no columns are allowed to store NULL VALUES the following statement can be used.



```
CREATE TABLE IF NOT EXISTS newauthor
(aut_id varchar(8) NOT NULL,
aut_name varchar(50) NOT NULL,
country varchar(25) NOT NULL,
home_city varchar(25) NOT NULL );
```

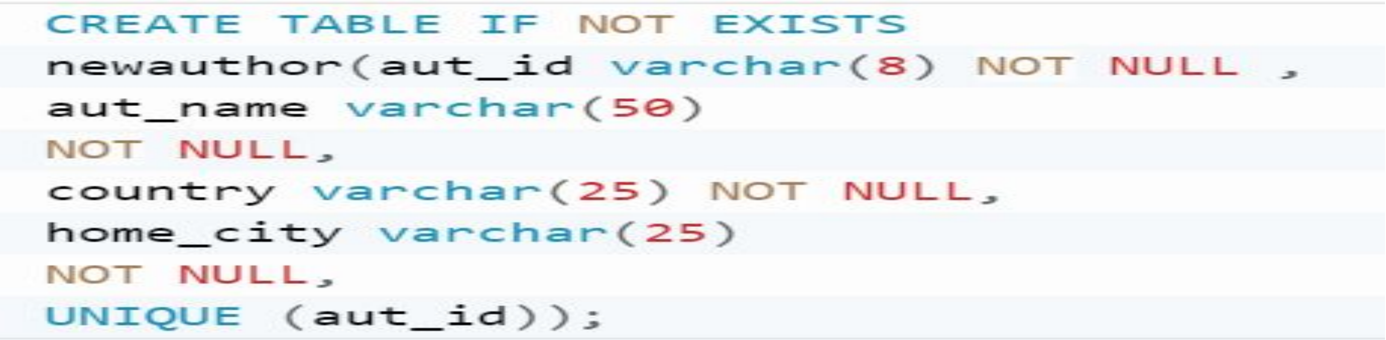
The f



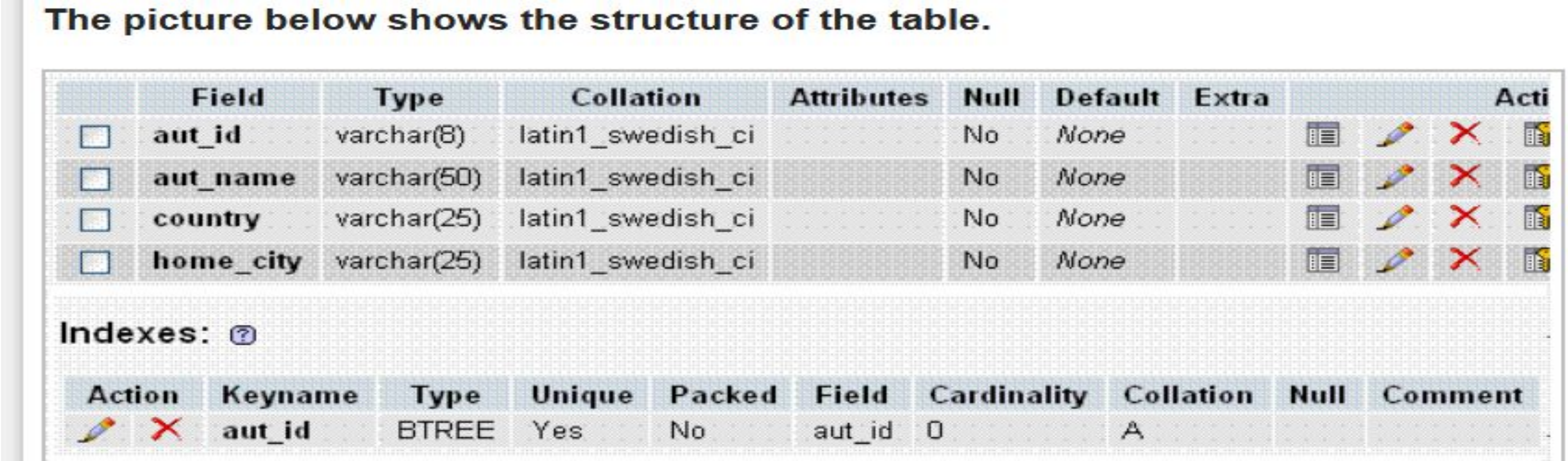
	Field	Type	Collation	Attributes	Null	Default	Extra
□	**aut_id**	varchar(8)	latin1_swedish_ci		No	*None*	
□	**aut_name**	varchar(50)	latin1_swedish_ci		No	*None*	
□	**country**	varchar(25)	latin1_swedish_ci		No	*None*	
□	**home_city**	varchar(25)	latin1_swedish_ci		No	*None*	

MySQL UNIQUE CONSTRAINT

- **Example**



```
CREATE TABLE IF NOT EXISTS
newauthor(aut_id varchar(8) NOT NULL ,
aut_name varchar(50)
NOT NULL,
country varchar(25) NOT NULL,
home_city varchar(25)
NOT NULL,
UNIQUE (aut_id));
```



The picture below shows the structure of the table.

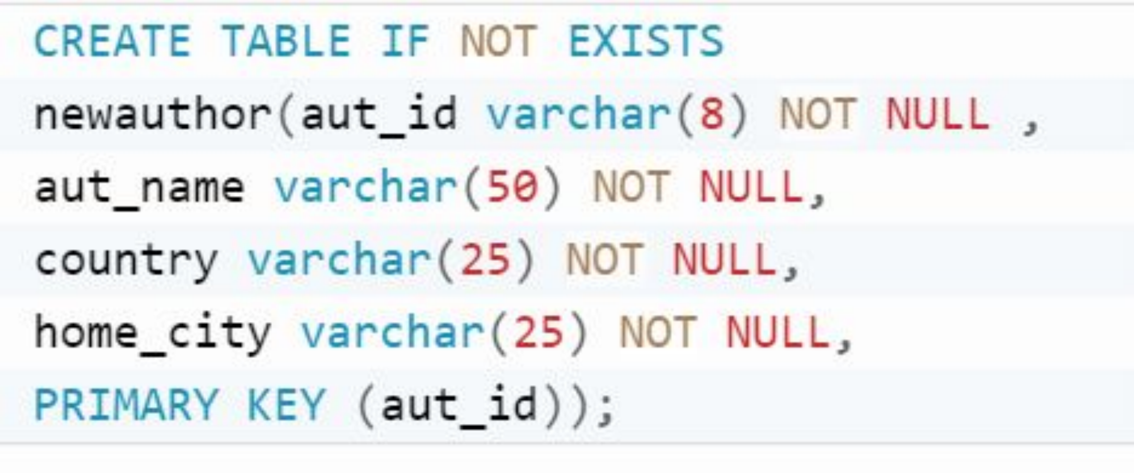
	Field	Type	Collation	Attributes	Null	Default	Extra	Acti
□	aut_id	varchar(8)	latin1_swedish_ci		No	*None*		▤ ✎ ✗ ▤
□	aut_name	varchar(50)	latin1_swedish_ci		No	*None*		▤ ✎ ✗ ▤
□	country	varchar(25)	latin1_swedish_ci		No	*None*		▤ ✎ ✗ ▤
□	home_city	varchar(25)	latin1_swedish_ci		No	*None*		▤ ✎ ✗ ▤

Indexes: ?

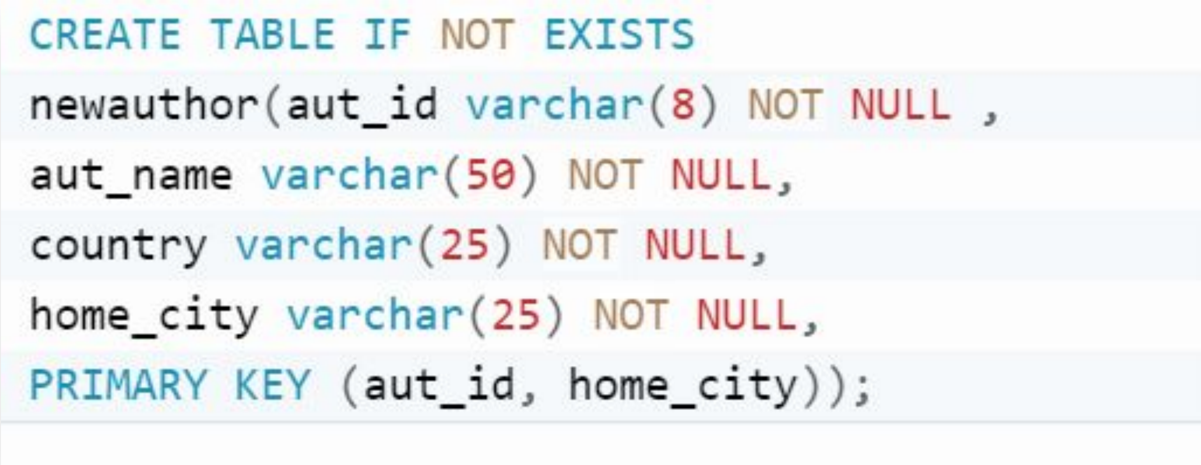
Action		Keyname	Type	Unique	Packed	Field	Cardinality	Collation	Null	Comment
✎	✗	aut_id	BTREE	Yes	No	aut_id	0	A		

MySQL PRIMARY KEY CONSTRAINT on Single and Multiple Columns

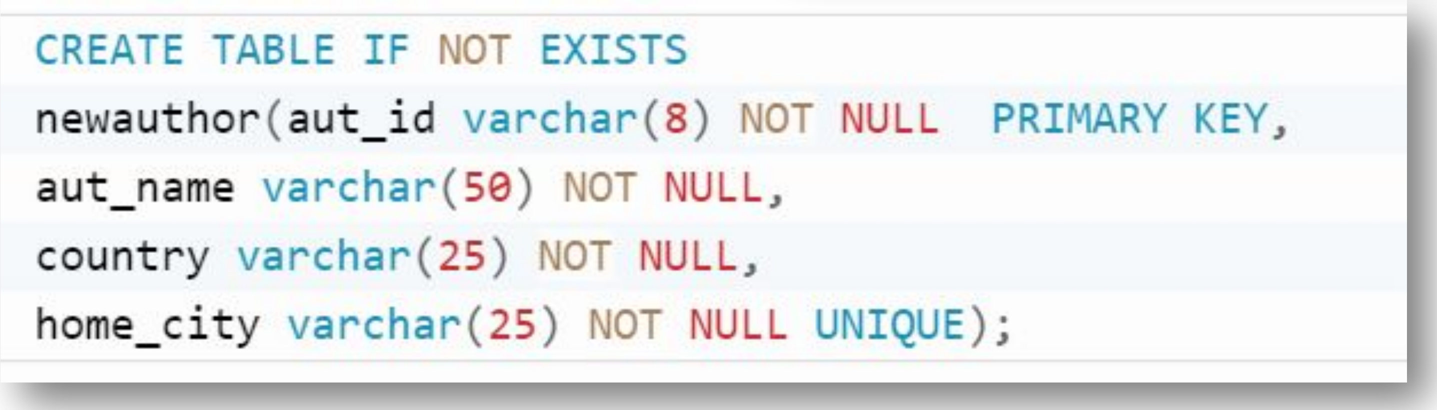
- Example



```
CREATE TABLE IF NOT EXISTS
newauthor(aut_id varchar(8) NOT NULL ,
aut_name varchar(50) NOT NULL,
country varchar(25) NOT NULL,
home_city varchar(25) NOT NULL,
PRIMARY KEY (aut_id));
```

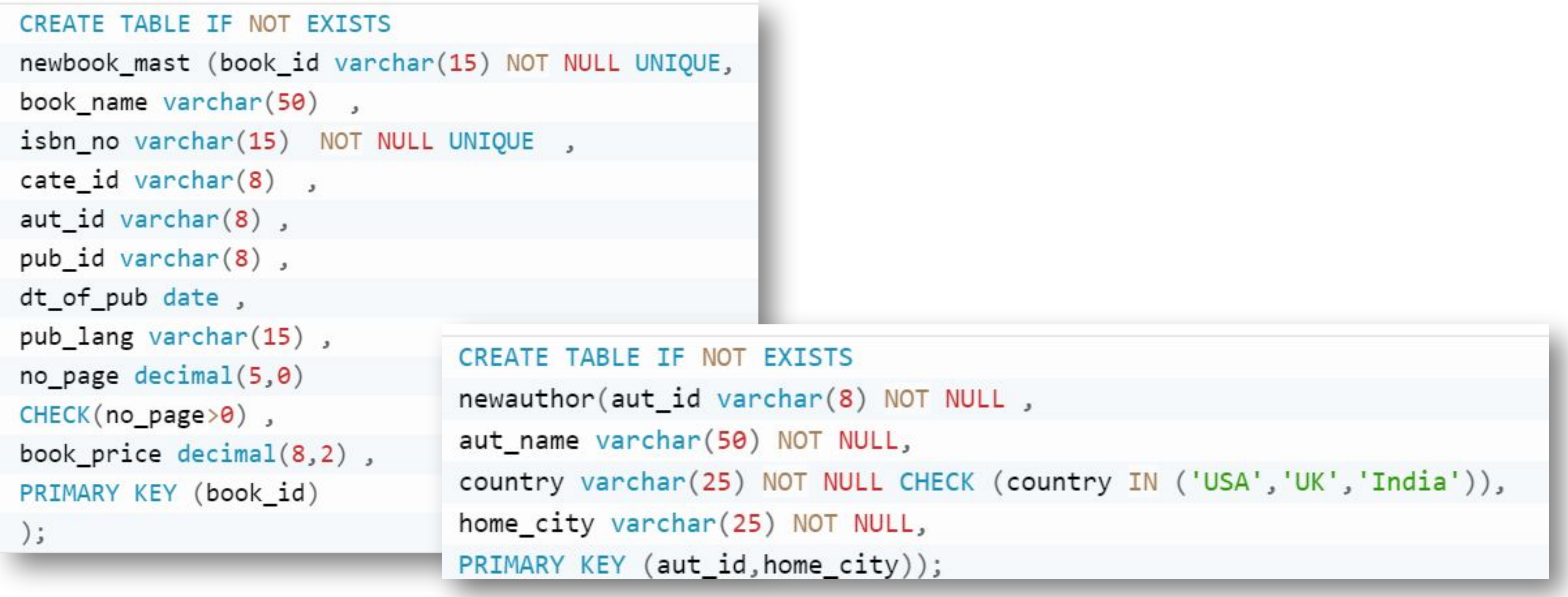


```
CREATE TABLE IF NOT EXISTS
newauthor(aut_id varchar(8) NOT NULL ,
aut_name varchar(50) NOT NULL,
country varchar(25) NOT NULL,
home_city varchar(25) NOT NULL,
PRIMARY KEY (aut_id, home_city));
```



```
CREATE TABLE IF NOT EXISTS
newauthor(aut_id varchar(8) NOT NULL  PRIMARY KEY,
aut_name varchar(50) NOT NULL,
country varchar(25) NOT NULL,
home_city varchar(25) NOT NULL UNIQUE);
```

MySQL CREATE TABLE to check values with CHECK CONSTRAINT using different operators

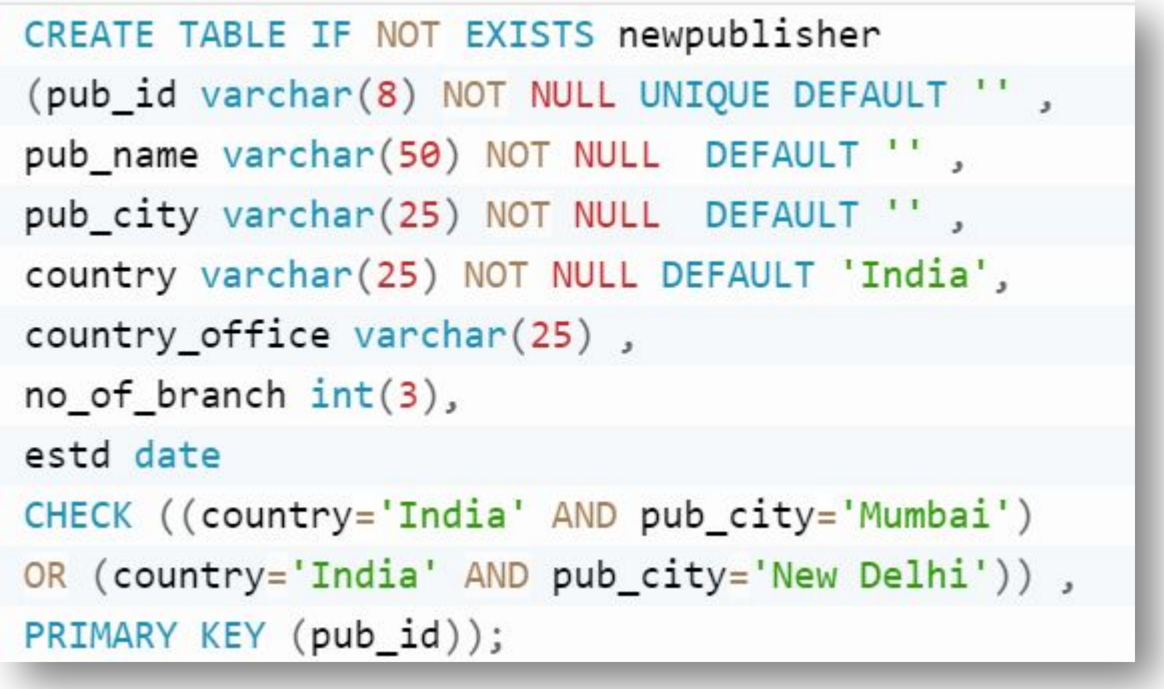


```
CREATE TABLE IF NOT EXISTS
newbook_mast (book_id varchar(15) NOT NULL UNIQUE,
book_name varchar(50)  ,
isbn_no varchar(15)  NOT NULL UNIQUE  ,
cate_id varchar(8)  ,
aut_id varchar(8) ,
pub_id varchar(8) ,
dt_of_pub date ,
pub_lang varchar(15) ,
no_page decimal(5,0)
CHECK(no_page>0) ,
book_price decimal(8,2) ,
PRIMARY KEY (book_id)
);
```

```
CREATE TABLE IF NOT EXISTS
newauthor(aut_id varchar(8) NOT NULL ,
aut_name varchar(50) NOT NULL,
country varchar(25) NOT NULL CHECK (country IN ('USA','UK','India')),
home_city varchar(25) NOT NULL,
PRIMARY KEY (aut_id,home_city));
```

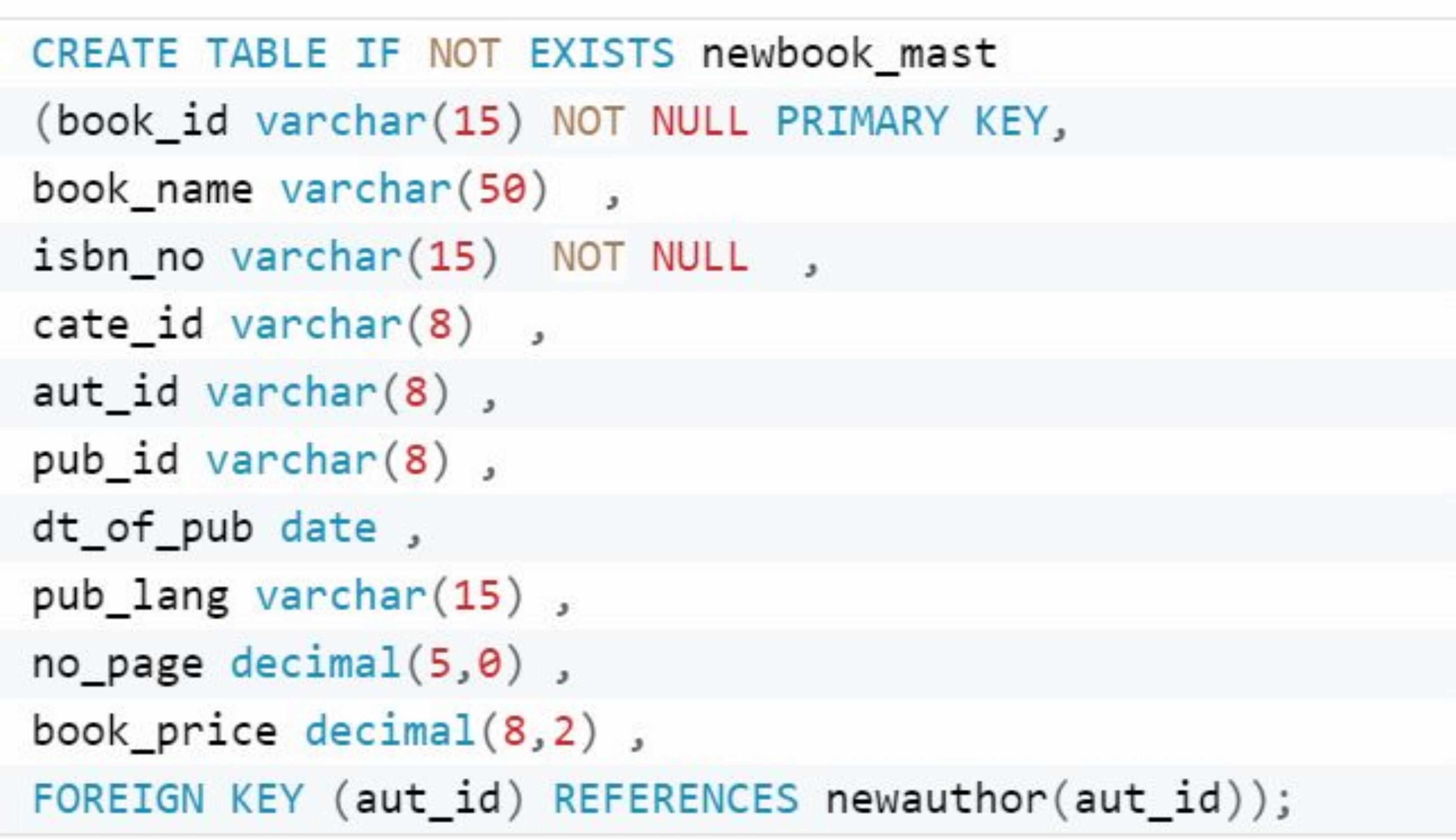
MySQL CREATE TABLE with DEFAULT CONSTRAINT

- **Example**



```
CREATE TABLE IF NOT EXISTS newpublisher
(pub_id varchar(8) NOT NULL UNIQUE DEFAULT '' ,
pub_name varchar(50) NOT NULL  DEFAULT '' ,
pub_city varchar(25) NOT NULL  DEFAULT '' ,
country varchar(25) NOT NULL DEFAULT 'India',
country_office varchar(25) ,
no_of_branch int(3),
estd date
CHECK ((country='India' AND pub_city='Mumbai')
OR (country='India' AND pub_city='New Delhi')) ,
PRIMARY KEY (pub_id));
```

MySQL creating table with FOREIGN KEY CONSTRAINT on single ,multiple column and multiple tables



```
CREATE TABLE IF NOT EXISTS newbook_mast
(book_id varchar(15) NOT NULL PRIMARY KEY,
book_name varchar(50)  ,
isbn_no varchar(15)  NOT NULL  ,
cate_id varchar(8)  ,
aut_id varchar(8) ,
pub_id varchar(8) ,
dt_of_pub date ,
pub_lang varchar(15) ,
no_page decimal(5,0) ,
book_price decimal(8,2) ,
FOREIGN KEY (aut_id) REFERENCES newauthor(aut_id));
```

MySQL: ALTER TABLE Statement

- **ALTER TABLE - ADD Column**
  - **Adding column(s) to a table**

  ALTER TABLE table_name ADD column_name column-definition;

  - **For example:**

    ALTER TABLE supplier ADD supplier_name varchar2(50);

  Add multiple columns in table

  ALTER TABLE contacts ADD last_name varchar(40) NOT NULL AFTER contact_id,
  ADD first_name varchar(35) NULL AFTER last_name;

- **ALTER TABLE - Drop column(s) in a table**

  ALTER TABLE table_name DROP COLUMN column_name;

  For example:

  ALTER TABLE supplier DROP COLUMN supplier_name;

MySQL: ALTER TABLE Statement

- **ALTER TABLE command to add a NOT NULL constraint**

```
ALTER TABLE table_name MODIFY column_name datatype NOT NULL;
```

- **ADD UNIQUE CONSTRAINT** to a table

```
ALTER TABLE Persons ADD UNIQUE (ID);

ALTER TABLE Persons ADD CONSTRAINT UC_Person UNIQUE
(ID,LastName);
```

- **DROP a UNIQUE Constraint**

```
ALTER TABLE Persons DROP INDEX UC_Person;
```

MySQL: ALTER TABLE Statement

Modify column in table

ALTER TABLE table_name MODIFY column_name column_definition [FIRST | AFTER column_name];

Example

ALTER TABLE contacts MODIFY last_name varchar(55) NULL AFTER contact_type, MODIFY first_name varchar(30) NOT NULL;

MySQL: ALTER TABLE Statement

SQL DEFAULT on ALTER TABLE

```
ALTER TABLE tablename
ALTER columnname SET DEFAULT colvalue;
```

Example

```
ALTER TABLE Persons
ALTER City SET DEFAULT ‘Pune';
```

DROP a DEFAULT Constraint

```
ALTER TABLE Persons
ALTER City DROP DEFAULT;
```

MySQL: ALTER TABLE Statement

Rename column in table
ALTER TABLE table_name CHANGE COLUMN old_name new_name column_definition [FIRST | AFTER column_name]

Example
ALTER TABLE contacts CHANGE COLUMN contact_type ctype varchar(20) NOT NULL;

Rename table
ALTER TABLE table_name RENAME TO new_table_name;
ALTER TABLE contacts RENAME TO people;

DCL STATEMENTS OF SQL

ACCESS CONTROL Statements SQL

- SQL supports discretionary data access control through
- the following statements:
  - GRANT
  - REVOKE
- GRANT: The statement is used to give privileges to other
- users.
- REVOKE: The statement is used to take away already
- given privileges from other users.

Steps

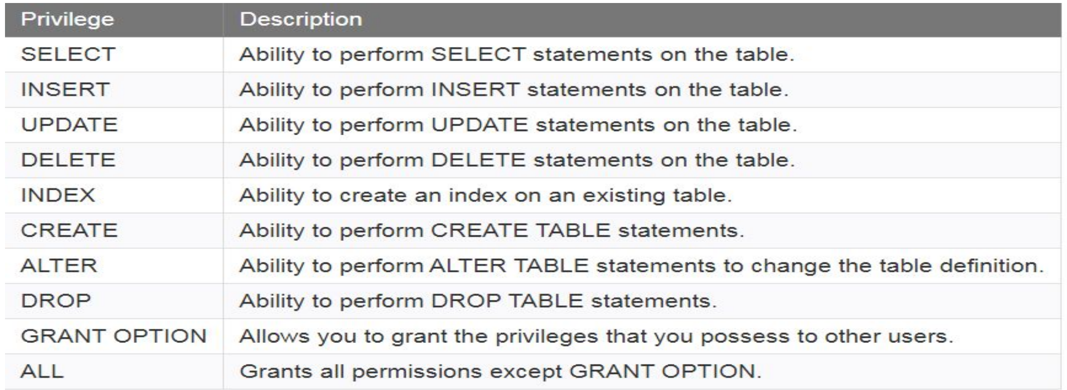
1. Login as root user
2. Create users
3. Give permissions using GRANT
4. Revoke permissions when needed using Revoke

Grant Privileges on Table

Syntax

GRANT privileges ON object TO user;

- **privileges**



Privilege	Description
SELECT	Ability to perform SELECT statements on the table.
INSERT	Ability to perform INSERT statements on the table.
UPDATE	Ability to perform UPDATE statements on the table.
DELETE	Ability to perform DELETE statements on the table.
INDEX	Ability to create an index on an existing table.
CREATE	Ability to perform CREATE TABLE statements.
ALTER	Ability to perform ALTER TABLE statements to change the table definition.
DROP	Ability to perform DROP TABLE statements.
GRANT OPTION	Allows you to grant the privileges that you possess to other users.
ALL	Grants all permissions except GRANT OPTION.

- **Object**
  - The name of the database object that you are granting permissions for. In the case of granting privileges on a table, this would be the table name.
- **User**
  - The name of the user that will be granted these privileges.

Example

```
GRANT SELECT, INSERT, UPDATE, DELETE ON contacts TO
'smithj'@'localhost';
```

```
GRANT SELECT ON contacts TO 'smithj'@'localhost';
```

```
GRANT All ON contacts TO 'smithj'@'localhost';
```

Revoke Privileges on Table

Syntax

```
REVOKE privileges ON object FROM user;
```

```
REVOKE Delete,Update ON contacts from 'smithj'@'localhost';
```

```
REVOKE select ON contacts TO ‘*'@'localhost';
```

```
REVOKE All ON contacts TO 'smithj'@'localhost';
```

Exercises -Batch A

Perform DDL and DCL commands

- Create a table with all constraints ,truncate a table, drop a table ,alter table by adding a column , dropping a column, adding and dropping different constraints, modify data type and length of column, rename table , column , create a user, grant different privileges and revoke

Exercises

- **Create a database which consist of the following tables with appropriate constraints like primary key, foreign key, check constrains, not null etc.**
  - Suppliers - S (S#, Name, Status, City)
  - Parts - P (P#, Pname, Colour, Weight, City)
  - Projects - J (J#, Jname, City)
  - Shipment - SPJ (S#, P#, J#, Qty)
- **Create a database which consist of the following tables with appropriate constraints like primary key, foreign key, check constrains, not null etc.**
  - employee (employee name, street, city) ,employee name is primary key
  - works (employee name, company name, salary)
  - company (company name, city) ,company name is primary key
  - manages (employee name, manager name)

Exercises -Batch B

Perform DDL and DCL commands

- Create a table with all constraints ,truncate a table, drop a table ,alter table by adding a column , dropping a column, adding and dropping different constraints, modify data type and length of column, , rename table , column , create a user, grant different privileges and revoke

Exercises

- **Create a database which consist of the following tables with appropriate constraints like primary key, foreign key, check constrains, not null etc.**
  - Hotel (HotelNo, Name, City) HotelNo is the primary key
  - Room (RoomNo, HotelNo, Type, Price)
  - Booking (HotelNo, GuestNo, DateFrom, DateTo, RoomNo)
  - Guest (GuestNo, GuestName, GuestAddress) GuestNo **is primary key**
- **Create a database which consist of the following tables with appropriate constraints like primary key, foreign key, check constrains, not null etc.**
  - emp (eno,ename,Zip,hdate)
  - parts(pno,pname,qty_on_hand, price)
  - customer(cno,cname,street,Zip,phone)
  - order(ono,cno,receivedate,shippeddate)
  - odetails(ono,pno,qty)
  - zipcode(Zip,city)

Exercises -Batch C

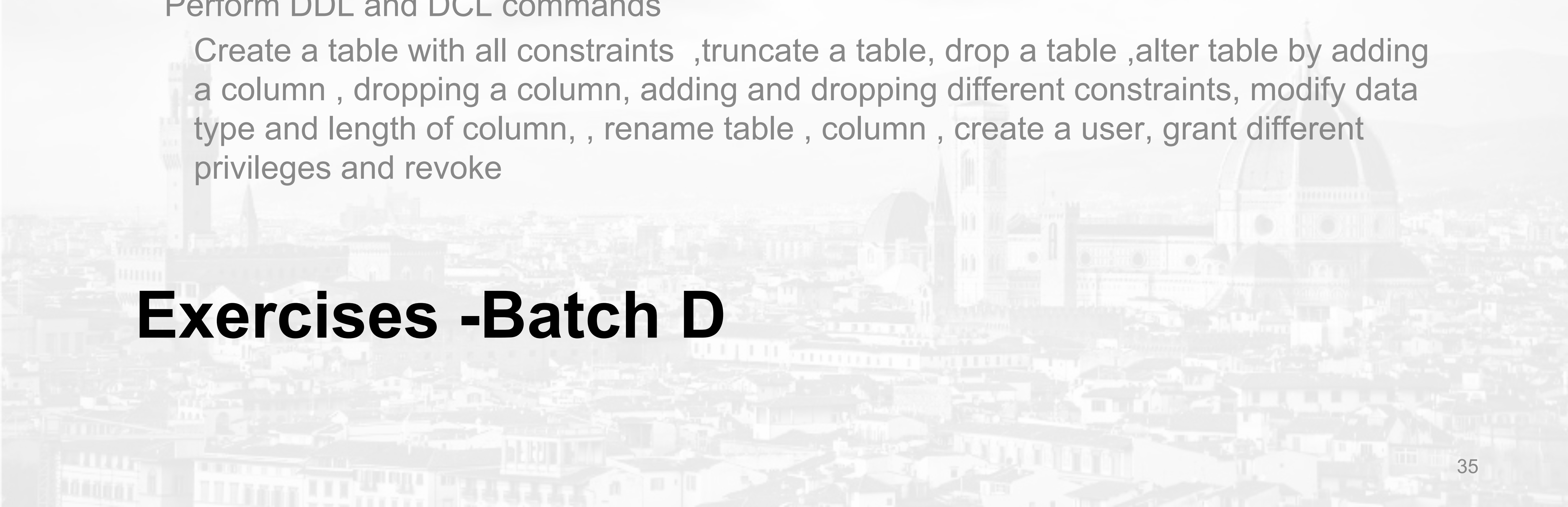
Perform DDL and DCL commands

- Create a table with all constraints ,truncate a table, drop a table ,alter table by adding a column , dropping a column, adding and dropping different constraints, modify data type and length of column, , rename table , column , create a user, grant different privileges and revoke

Exercises

- **Create a database which consist of the following tables with appropriate constraints like primary key, foreign key, check constrains, not null etc.**
  - Project(project_id,proj_name,chief_arch) project_id is primary key
  - Employee(Emp_id,Emp_name) Emp_id is primary key
  - Assigned-To(Project_id,Emp_id)

- **Create a database which consist of the following tables with appropriate constraints like primary key, foreign key, check constrains, not null etc.**
  - Employee(emp_no,name,skill,pay-rate) eno primary key
  - Position(posting_no,skill) posting_no primary key
  - Duty_allocation(posting_no,emp_no,day,shift)



Exercises -Batch D

Perform DDL and DCL commands

Create a table with all constraints ,truncate a table, drop a table ,alter table by adding a column , dropping a column, adding and dropping different constraints, modify data type and length of column, , rename table , column , create a user, grant different privileges and revoke

Exercises

- **Create a database which consist of the following tables with appropriate constraints like primary key, foreign key, check constrains, not null etc.**
  - Account(Acc_no, branch_name,balance)
  - branch(branch_name,branch_city,assets)
  - customer(cust_name,cust_street,cust_city)
  - Depositor(cust_name,acc_no)
  - Loan(loan_no,branch_name,amount)
  - Borrower(cust_name,loan_no)

- **Create a database which consist of the following tables with appropriate constraints like primary key, foreign key, check constrains, not null etc.**
  - Emp (eno,ename, job,hiredate,salary,commission,deptno,)
  - dept(deptno,deptname,location)

    eno is primary key in emp, deptno is primary key in dept

Perform DDL and DCL commands

Create a table with all constraints ,truncate a table, drop a table ,alter table by adding a column , dropping a column, adding and dropping different constraints, modify data type and length of column, , rename table , column , create a user, grant different privileges and revoke

Exercises -Batch E

Exercises

Schema of the university database

classroom(building, room number, capacity)

department(dept name, building, budget)

course(course id, title, dept name, credits)

instructor(ID, name, dept name, salary)

section(course id, sec id, semester, year, building, room number, time slot id)

teaches(ID, course id, sec id, semester, year)

student(ID, name, dept name, tot cred)

takes(ID, course id, sec id, semester, year, grade)

advisor(s ID, i ID)

time slot(time slot id, day, start time, end time)

prereq(course id, prereq id)

Thank You!