1. To use mysql

mysql -u root -p then press enter (password)

2. To get mysql version on ubuntu

select version(); (on mysql command prompt)
mysql –version or mysql -V on ubuntu shell prompt
lsb_release -a (for ubuntu release ; linux standard base info)

3. To get info about all databases;
show databases;

4. start working on a database
use mysql;

5. All tables within a used database

show tables;

5. to see scheme of a table
describe tablename; (column listing)

show create table tablename; (create statement)

6. To get current date

select curdate();
select year(now());

To get help on any literal, use:
help ‘create’
help ‘data types’ => all permissible data types in mysql

To get information about database/table:
show databases;
show tables;
show create table t1; => command to create table
show table status ; all information maintained in info schema like total rows, table size
desc table /G; => gives result column-wise i.e. vertically for each column

7. Creating a table:

requirement: table scheme with datatype of columns
constraints on table

Date,time and YEAR data types are stored as sequence of characters

date: 'YYYY-MM-DD'
time: hh:mm:ss
Year : 1-byte data type with values 0 to 255 mapped to range: 1901 to 2155
if only two digits of a year are given then 00-69 is stored as 2000 to 2069; 70-99 is taken as 1970 to 1999

Problem statement:

College maintains data about students registered in different societies with the objective of availability of information as per requirement. E.g.

Total students registered in all societies.
Total students in each society? Popular society? Least popular society? Popularity each year?? list of students registered in a society?? which course students are opting 'society A' in majority??

Which tables need to be maintained and why??

STUDENT-SOCIETY Database

Database Scheme for student-society Database

STUDENT	Rollno	Name	Course	DOB
	Char(6)	varchar(20)	varchar(15)	Date

SOCIETY	SID Char(6)	SName varchar(20)	Mentor varchar(20)	Total seats Unsigned int

ENROLLMENT	Rollno	SID	Dateof-enrollment

AD-1201	S1	2023/01/06
AD-1201	S2	
AD-1202	S2	

Note: underlined attributes are primary keys

ENROLLMENT(Rollno) is foreign key referring to STUDENT(Rollno)

ENROLLMENT(SID) is foreign key referring to SOCIETY(SID)

Other constraints:

Name of society and student cannot be NULL. By default, total seats in each society is 10

Primary key imposes key constraint and Entity integrity constraint
Foreign key imposes Referential integrity constraint.

Syntax for creating a table

CREATE TABLE table_name
(column_name1 column_type NULL/Not NULL default value,
column_name2 column_type, ...
constraint constname definition);

create table student(rollno char(6), name varchar(20) not null, cname varchar(10), dob date);
alter table student add constraint pk1 primary key(rollno);

or we may write it as:

create table student(rollno char(6), name varchar(20) not null, cname varchar(10), dob date not null default '2000-03-01', constraint pk1 primary key(rollno));

create table society(sid char(6), sname varchar(20) not null, mentorname varchar(20) not null , capacity int unsigned default 50);

alter table society add constraint pk2 primary key(sid);

Use enum() data type to restrict values of a field to a set of values: for example change course type as:

alter table student modify course enum('Cs(hons)','PSCS','Bcom(h)');

Primary key is constraint is **dropped using the following in mysql**:

alter table society drop primary key;
alter table society drop constraint pk2;

Primary key constraint is **dropped using the following in oracle**:
alter table society drop pk2; #using constraint name

Note: Never add foreign key in table if its corresponding primary key is not generated.

Points to remember while creating database:

1. Start with creation of table with no foreign key.
2. If no such table, then start with table having minimum foreign keys.
   I. In creation, donot specify foreign key constraint as reference table has not created so far.
   II. After creating referenced table, alter the former table and add foreign key constraints.
3. **Populate tables with data once all tables are created to avoid ant types of errors in data entry.**

create table enrollment(rollno char(6), sid char(6), constraint pk3 primary key(rollno,sid));

alter table enrollment add constraint fk1 foreign key(sid) references society(sid);

alter table enrollment add constraint fk2 foreign key(RollNo) references student(RollNo);

dropping foreign key in mysql
alter table enrollment drop foreign key fk2;

dropping foreign key in oracle
alter table enrollment drop constraint fk2;

Alternatively, we may write if referenced keys are defined:

create table enrollment(rollno char(6), sid char(6), constraint pk3 primary key(rollno,sid), constraint fk1 foreign key(sid) references society(sid), constraint fk2 foreign key(sid) references society(sid));

****to change datatypes/default/ of existing attributes**
****drops default value from the table scheme**

alter table society alter capacity drop default;

#changing attribute name

alter table society change sname socname varchar(20);

alter table t1 change column f1 ffff decimal(10,1);

#change tablename
ALTER TABLE society RENAME TO Society;

insert data in tables: example...
insert into society values ('s2','writing skills',12,'AAA'); .....
insert into student values('12','xxxxx','cs hons','2012-06-25'); ....

Removing a table:
Drop a table
Drop student;

Delete a record

Delete from table where condition;

Updating records in table

Update table set values,... where condition;
update society set mentorname ='Dr. manoj' where sname like 'deba%';

Dropping default constraint from a field
alter table society alter capacity drop default;

#for setting defualt values
alter table STUDENT alter SName set default 'XXX';

#for removing NULL/Not NULL
alter table STUDENT modify SDateOfBirth date NOT NULL;

#looks for space in a fieldvalue

select * from person where locate(' ',pname) is TRUE;
select * from person where position(' ' IN pname) is TRUE;

Function	Description
ASCII	Returns the number code that represents the specific character
CHAR_LENGTH	Returns the length of the specified string (in characters)
CHARACTER_LENGTH	Returns the length of the specified string (in characters)
CONCAT	Concatenates two or more expressions together
CONCAT_WS	Concatenates two or more expressions together and adds a separator between them
FIELD	Returns the position of a value in a list of values
FIND_IN_SET	Returns the position of a string in a string list
FORMAT	Formats a number as a format of "#,###.##", rounding it to a certain number of decimal places
INSERT	Inserts a substring into a string at a specified position for a certain number of characters
INSTR	Returns the position of the first occurrence of a string in another string
LCASE	Converts a string to lower-case

LEFT	Extracts a substring from a string (starting from left)
LENGTH	Returns the length of the specified string (in bytes)
LOCATE	Returns the position of the first occurrence of a substring in a string
LOWER	Converts a string to lower-case
LPAD	Returns a string that is left-padded with a specified string to a certain length
LTRIM	Removes leading spaces from a string
MID	Extracts a substring from a string (starting at any position)
POSITION	Returns the position of the first occurrence of a substring in a string
REPEAT	Repeats a string a specified number of times
REPLACE	Replaces all occurrences of a specified string
REVERSE	Reverses a string and returns the result
RIGHT	Extracts a substring from a string (starting from right)
RPAD	Returns a string that is right-padded with a specified string to a certain length
RTRIM	Removes trailing spaces from a string
SPACE	Returns a string with a specified number of spaces
STRCMP	Tests whether two strings are the same
SUBSTR	Extracts a substring from a string (starting at any position)
SUBSTRING	Extracts a substring from a string (starting at any position)
SUBSTRING_INDEX	Returns the substring of *string* before *number* of occurrences of *delimiter*
TRIM	Removes leading and trailing spaces from a string
UCASE	Converts a string to upper-case
UPPER	Converts a string to upper-case

To temporary change attribute names in a query, it is not possible to write new names with table as mentioned in book, but alternative is as follows:

```
select * from society join (select Srollno as R, Id as sid, DOE from enroll) E on society.sid=E.sid;
```

Natural Joining two tables:

```
select * from society natural join (select Srollno as R, Id as sid, DOE from enroll) NEW;
```

Derived table must be given a name as shown above.

Queries to be done as part of first practical exercise:

1. Retrieve names of student enrolled in any society

2. Retrieve all society names
3. Retrieve students names starting with letter ‘A’
4. Retrieve students studying in course ‘computer sc’ or ‘chemistry’
5. Retrieve students whose rollno either starts with ‘X’ or ‘Z’
6. Find society whose capacity is more than 10
7. Update society table for mentor name for a specific society
8. Remove details for a student who is not enrolled in any society
9. Increment capacity of each society by 10%

10. Find the student names who are not enrolled in any society

11. Find total number of students whose age is > 20 years

12 Find the student names enrolled in atleast two societies
13. Find society names in which any student is enrolled
14 Find names of all students enrolled in any society and society names in which any student is enrolled
15. Find names of students who are enrolled in all three societies ‘debating’, ‘dancing’ and ‘sashakt’.
16. Find society names that has ‘abc’ as mentor or ‘abc’ as the name of enrolled student.
17. Find society names whose mentor name is same as that of any enrolled student in it.
18. Find the society names in which number of enrolled students are less than its capacity.
19. Display the vacant seats for each society.
20. Find society names in which more than five students have enrolled in the given year
21. Add enrolment fees paid (‘yes’/’No’) field in the enrollment table.
22. Update date of enrolment of society s1 to ‘2018-01-15’, s2 to current date and s3 to ‘2018-01-02’.
23. Find society names whose enrolment is over.
24. Find common societies of students of courses ‘cs(hons)’ and ‘pscs’
25. Create a view to keep track of society names with total number of students enrolled in it.
26. Find student names enrolled in all societies.
27. Count societies with student enrolled > 3
28 add column contact in student with default value
29. Find the name of oldest and youngest student in class along with their age and DOB
30. Find the most popular and least popular society name (on the basis of enrolled students)

31. Find names of students born in year 2001 and enrolled in atleast one society
32 Remove default value of any field
33 Find society names where students have enrolled in month Jan/Feb
34 Display all students details alongwith society name if they are enrolled in any society
35 Display society names in uppercase and padded with character * to get a length of 10 characters which are mentored by mentors whose names start with 'M' and ends with 'r' and capacity is between 10 to 20.

Note: the submitted document (pdf) must include student name, rollnumber, course, semester and date of submission and must consists of
database scheme, DDL statements (create/alter/drop), DML statements (few insert statements and update statements) and all the retrieval queries.