

Structured Query Language

- best practices

SQL

SQL is a fourth generation language used for organizing, management and fetching of data from relational database.

RDBMS like Oracle, IBM DB2, Sybase, MySQL and Microsoft SQL Server use SQL as query language for their database.

SQL can be divided into three groups of commands, depending on their purpose.

- Data Definition Language (DDL).
- Data Manipulation Language (DML) and Data Query Language (DQL).
- Data Control Language (DCL)

Data Definition Language :

Data Definition Language is a part of SQL that is responsible for the creation, alteration and dropping of tables and other database objects like views, indexes etc.

Examples - CREATE, ALTER, DROP, TRUNCATE

Data Manipulation Language and Data Query Language

Data manipulation commands (insert, delete, update) manipulate data.
The DQL includes the select commands that run queries and fetches the data.

DML :	UPDATE	DELETE	INSERT
DQL :	SELECT		

Data Control Language :

The commands of data control language are related to the security of the database and perform tasks of granting or revoking privileges so users can access certain objects in the database. It also use the commands for management of transactional statements by successfully committing it or rollback.

The DCL commands are :

GRANT

REVOKE

COMMIT

ROLLBACK

Exercise

1. Create a new database

create database MyDb;

2. Get connected with database

use MyDb;

3. To show all tables in database

show Tables;

Data type

The data type is a keyword that set a guideline for SQL to understand what type of data is expected inside each column.

In MySQL there are three main data types: string, numeric, and date and time.

String– char(size), varchar(size), binary(size), varbinary(size), Text(size), BLOB(size)

Numeric – BIT(size), SMALLINT, INT, BIGINT, FLOAT, DECIMAL(size, precision)

Date and Time – Date, Time, TimeStamp, Year

Exercise

4. To create new table Country

create table Country (

CountryCode int primary key,

// column level constraint declaration

CountryName varchar(30)

);

5. To view table structure with column details

describe Country;

Exercise

6. To add row of data in Country table

Insert into Country(CountryCode,CountryName)values(91,'India');

Insert into Country values(1,'USA');

**Insert into Country(CountryCode,CountryName)
values(44,'UK'),(33,'France'),(7,'Russia'),(49,'Germany'),(86,'China');**

7. To view all data in Country table

Select * from Country;

Select CountryCode, CountryName from Country;

Exercise

8. To create another table “CountryCopy”, similar in structure like Country.

create table CountryCopy like Country;

9. To copy all records from Country into CountryCopy

Insert into CountryCopy Select * from Country;

(view details)

Constraint

SQL constraints are used to specify rules for data in a table.

Constraints can be specified when the table is created with the CREATE TABLE statement, or after the table is created with the ALTER TABLE statement.

Constraint ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted.

The constraint that can be applied in SQL are-

primary key

foreign key

not null

default

check

unique

Exercise

10. To create a new table author

```
create table Author  
(  
  AuthorId int auto_increment,  
  AuthorName varchar(25),  
  DateOfBirth datetime,  
  CountryCode int,  
  constraint pk_authorId primary key(AuthorId),           // table level constraint declaration  
  constraint fk_country foreign key(iCountryCode) references Country(iCountryCode)  
);
```

Exercise

11. Modify "Author" Table by adding a new column Gender.

Alter table Author

Add Gender char(1);

12. Rename column DateOfBirth as BirthDate in Author table

Alter table Author

Rename column DateOfBirth to BirthDate;

13. Modify datatype size of Gender Column as char(6)

Alter table Author

Modify Gender char(6);

Exercise

14. Insertion of records in Author Table

insert into Author(AuthorId, AuthorName, BirthDate, CountryCode, Gender) values

(default,'Mary Curie','1867-11-07',33, 'Female'), (default,'RK Narayanan','1906-10-10',91,'Male'),

(default,'Rabindranath Tagore','1861-05-07',91,'Male'),

(default,'Leo Tolstoy','1828-09-09',7,'Male'),

(default,'Mahadevi Verma','1907-03-26',91,'Female'),

(default,'Lewis Carol','1832-01-27',44,'Male'),

(default,'Ernest Hemingway','1899-07-21',1,'Male'),

(default,'Charles Dickens','1812-02-07',44,'Male'),

(default,'Dr APJ Kalam','1931-10-15',91,'Male');

Exercise

15. To modify author name “R.K.Narayanan” to “R.K.Narayan” (AuthorId = 2)

Update Author

Set AuthorName = 'R.K.Narayan'

where AuthorId = 2

Syntax-

Update TableName

Set ColumnName1 = value, ColumnName2 = value, ...

where KeyColumnName = searchvalue

Exercise

16. To delete a record from “CountryCopy” table where CountryCode is 86.

Delete from CountryCopy where CountryCode = 86;

17. To remove all record from “CountryCopy” without any condition

Truncate table CountryCopy;

Exercise

18. Create a new table Book with applicable constraint like unique, check, foreign key (column level)

create table Book

(

BookCode int auto_increment,

Title varchar(35),

Category varchar(25),

AuthorId int references Author(AuthorId),

Price decimal(6,2),

TotalPages int,

constraint pkBookCode primary key(BookCode),

constraint unqTitle unique>Title),

constraint chkPrice check(Price > 0)

);

Exercise

19. Insert records in
Book Table

Insert into Book(BookCode,Title,Category,AuthorId,Price,TotalPages)
VALUES

(default,'Radioactivity','Science',1,758.75,1837),

(default,'Wings of Fire','Science',9,1150,2956),

(default,'Meri Pariwar','Novel',5,235,316),

(default,'Malgudy Days','Stories',2,156,85),

(default,'Swami and Friends','Stories',2,116,70),

(default,'Godan','Novel',null,160,336),

(default,'One More Over','Sports',null,38,63),

(default,'Kabuliwala','Stories',3,68,138),

(default,'Geetanjali','Literature',3,335,2238),

(default,'The Waterfall','Poetry',3,25,32),

(default,'The Common Man','Stories',2,200,275),

(default,'War and Peace','Literature',4,556.5,788),

(default,'Alice in Wonderland','Tales',8,100,128);

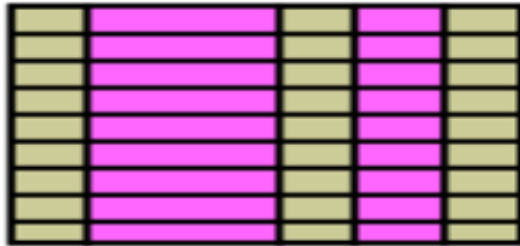
Select Statement

A SELECT statement retrieves information from the database. With a SELECT statement, we can use the following capabilities:

- **Projection:** Select the columns in a table that are returned by a query. Select as few or as many of the columns as required.
- **Selection:** Select the rows in a table that are returned by a query. Various criteria can be used to restrict the rows that are retrieved.
- **Joining:** Bring together data that is stored in different tables by specifying the link between them.

SQL SELECT Statements

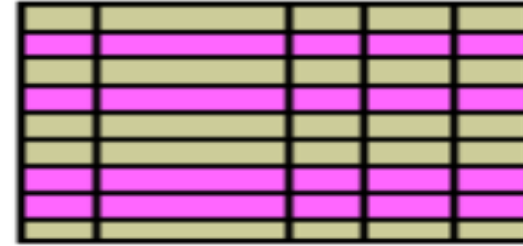
Projection



A diagram illustrating the Projection operation. It shows a 10x5 grid representing Table 1. The second, third, and fourth columns are highlighted in pink, indicating they are the selected attributes for the projection.

Table 1

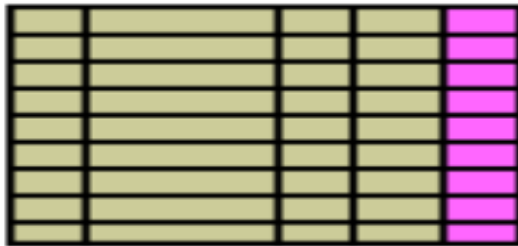
Selection



A diagram illustrating the Selection operation. It shows a 10x5 grid representing Table 1. The first, second, and third rows are highlighted in pink, indicating they are the selected rows based on a condition.

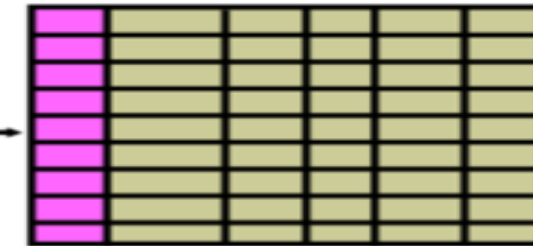
Table 1

Join



A diagram showing the result of a Join operation. It is a 10x5 grid where the first four columns are from Table 1 and the fifth column is from Table 2. The fifth column is highlighted in pink.

Table 1



A diagram of Table 2, which is a 10x5 grid. The first column is highlighted in pink.

Table 2

Exercise

20. Write a SQL statement that display the names of all authors and their respective birthdate.

Keep the column heading names as “Author Name” and “Date of Birth” respectively. The records must be sorted in ascending order based on name of the authors.

Author Name	Date of Birth
Charles Dickens	1812-02-07
Dr APJ Kalam	1931-10-15
Ernest Hemingway	1899-07-21
Leo Tolstoy	1828-09-09
Lewis Carol	1832-01-27
Mahadevi Verma	1907-03-26
Mary Curie	1867-11-07
R.K.Narayan	1906-10-10
Rabindranath Tagore	1861-05-07

**Select AuthorName as 'Author Name',
Date(BirthDate) as 'Date of Birth' from Author
order by AuthorName;**

Using the ORDER BY Clause

- Sort retrieved rows with the `ORDER BY` clause:
 - `ASC`: Ascending order, default
 - `DESC`: Descending order
- The `ORDER BY` clause comes last in the `SELECT` statement:

Exercise – Round(x, y)

21. Write a SQL statement that display the title of the book, price, discount calculated as 15 percent of price (restricted up to 2 decimal places) and actual offered price.

Keep the column heading names as “BookName”, “Price”, “Discount” and “OfferedPrice” respectively. The records must be sorted in descending order based price of the book.

BookName	Price	Discount	Offered Price
Wings of Fire	1150.00	172.50	977.50
Radioactivity	758.75	113.81	644.94
War and Peace	556.50	83.48	473.03
Geetanjali	335.00	50.25	284.75
Meri Pariwar	235.00	35.25	199.75
The Common Man	200.00	30.00	170.00
Godan	160.00	24.00	136.00
Malgudy Days	156.00	23.40	132.60
Swami and Friends	116.00	17.40	98.60
Alice in Wonderland	100.00	15.00	85.00
Kabuliwala	68.00	10.20	57.80
One More Over	38.00	5.70	32.30
The Waterfall	25.00	3.75	21.25

**Select Title 'BookName', Price 'Price',
Round(Price*0.15,2) 'Discount',
Round((Price - (Price*0.15)),2) 'Offered Price'
from Book order by Price desc**

Exercise – distinct, cartesian product

22. Write a SQL statement that display the name of the countries of every authors.

The records must be sorted in ascending order on the basis of CountryName

CountryName
France
India
Russia
UK
USA

**Select distinct CountryName from Author, Country
where Author.CountryCode = Country.CountryCode
order by CountryName;**

Generating a Cartesian Product

EMPLOYEES (20 rows)

	EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
1	100	King	90
2	101	Kochhar	90
3	102	De Haan	90
4	103	Hunold	60

...

19	205	Higgins	110
20	206	Gietz	110

DEPARTMENTS (8 rows)

	DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID
1	10	Administration	1700
2	20	Marketing	1800
3	50	Shipping	1500
4	60	IT	1400
5	80	Sales	2500
6	90	Executive	1700
7	110	Accounting	1700
8	190	Contracting	1700

Cartesian product:
20 x 8 = 160 rows

	EMPLOYEE_ID	DEPARTMENT_ID	LOCATION_ID
1	100	90	1700
2	101	90	1700
3	102	90	1700
4	103	60	1700

...

159	205	110	1700
160	206	110	1700

Creating Cross Joins

- The `CROSS JOIN` clause produces the cross-product of two tables.
- This is also called a Cartesian product between the two tables.

Exercise – Concat(), IfNull()

23. Write a SQL statement that display the column with heading “Book Details” as concatenated values of ‘AuthorId – Title’. If the value of AuthorId is unknown display as ‘N/A’.

The records must be sorted in ascending order on the basis of AuthorId.

Book Details
N/A-Godan
N/A-One More Over
1-Radioactivity
2-Malgudy Days
2-Swami and Friends
2-The Common Man
3-Kabuliwala
3-Geetanjali
3-The Waterfall
4-War and Peace
5-Meri Pariwar
8-Alice in Wonderland
9-Wings of Fire

**Select Concat(IfNull(AuthorId,'N/A'), '-', Title) 'Book Details'
from Book
order by AuthorId**

Exercise

24. Write a SQL statement that display the name of the author, title of the book and the name of the country. The records must be sorted in ascending order on the basis of AuthorName.

AuthorName	Title	CountryName
Charles Dickens	Alice in Wonderland	UK
Dr APJ Kalam	Wings of Fire	India
Leo Tolstoy	War and Peace	Russia
Mahadevi Verma	Meri Pariwar	India
Mary Curie	Radioactivity	France
R.K.Narayan	Malgudy Days	India
R.K.Narayan	Swami and Friends	India
R.K.Narayan	The Common Man	India
Rabindranath Tagore	Kabuliwala	India
Rabindranath Tagore	Geetanjali	India
Rabindranath Tagore	The Waterfall	India

**Select a.AuthorName, b.Title, c.CountryName
from Author a, Book b, Country c
where a.AuthorId = b.AuthorId
and a.CountryCode = c.CountryCode
order by AuthorName;**

Comparison Operators

Operator	Meaning
=	Equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
<>	Not equal to
BETWEEN ...AND...	Between two values (inclusive)
IN(set)	Match any of a list of values
LIKE	Match a character pattern
IS NULL	Is a null value

Snipping Tool

Exercises

25. Display the details of those books whose Price is less than 200.

Select * from Book where Price < 200

26. Display the details of those books whose title contains the string “days” in it.

Select * from Book where Title like '%Days%';

27. Display the AuthorName and BirthDate of those authors who were born in nineteenth century. The records must be sorted in ascending order based on BirthDate.

**Select AuthorName, Date(BirthDate) from Author
where BirthDate Between
'1800-01-01' And '1899-12-31'
order by BirthDate;**

AuthorName	Date(BirthDate)
Charles Dickens	1812-02-07
Leo Tolstoy	1828-09-09
Lewis Carol	1832-01-27
Rabindranath Tagore	1861-05-07
Mary Curie	1867-11-07
Ernest Hemingway	1899-07-21

Exercises – IN, LIKE, Wildcard character

28. Write a SQL Statement to select the Title of the Books written by British and French Author.

Title
Alice in Wonderland
Radioactivity

**Select Title from Book where AuthorId IN
(select AuthorId from Author where CountryCode IN
(Select CountryCode from Country where CountryName IN ('UK','France')))
order by Title;**

29. Write a SQL Statement to fetch the details of those books whose title contains letter 'a' as second character. Arrange the records in ascending order based on BookCode.

Select * from Book where Title Like '_a%' order by BookCode;

BookCode	Title	Category	AuthorId	Price	TotalPages
1	Radioactivity	Science	1	758.75	1837
4	Malgudy Days	Stories	2	156.00	85
8	Kabuliwala	Stories	3	68.00	138
12	War and Peace	Literature	4	556.50	788

Exercises

30. Write a SQL Statement to fetch the details of those authors who are not from India and USA.

Select * from Author where CountryCode NOT IN

(Select CountryCode from Country where CountryName IN ('India', 'USA'));

31. Write a SQL Statement to fetch the title and price of all expensive books whose prices are greater than the average price. Sort the records on the basis of Price in descending order.

Select Title 'Expensive Book', Price from Book where Price > (Select Avg(Price) from Book)
order by Price DESC;

Expensive Book	Price
Wings of Fire	1150.00
Radioactivity	758.75
War and Peace	556.50
Geetanjali	335.00

Exercises – Limit, Round(), DateDiff(), String functions

32. Write a SQL Statement to fetch the details of the oldest Author.

```
SELECT * FROM Author ORDER BY BirthDate LIMIT 1;
```

or

```
Select * from Author where Round(DateDiff(current_date(),Birthdate)/365,1) =  
(Select Max(Age) from (Select AuthorId, Round(DateDiff(current_date(),Birthdate)/365,1) as Age  
from Author) T);
```

AuthorId	AuthorName	BirthDate	CountryCode	Gender
8	Charles Dickens	1812-02-07 00:00:00	44	Male

33. Write a SQL Statement to display FirstName in uppercase and LastName in lowercase for all authors having names in AuthorName field.

```
Select upper(substr(AuthorName,1,locate(' ',  
AuthorName))) as FirstName,  
lower(substr(AuthorName,locate(' ',AuthorName))) as  
LastName from Author order by 1;
```

FirstName	LastName
CHARLES	dickens
DR	apj kalam
ERNEST	hemingway
LEO	tolstoy
LEWIS	carol
MAHADEVI	verma
MARY	curie
R.K.	narayan
RABINDRANATH	tagore

What Are Group Functions?

Group functions operate on sets of rows to give one result per group.

EMPLOYEES

	DEPARTMENT_ID	SALARY
1	90	24000
2	90	17000
3	90	17000
4	60	9000
5	60	6000
6	60	4200
7	50	5800
8	50	3500
9	50	3100
10	50	2600
...		
18	20	6000
19	110	12000
20	110	8300

**Maximum salary in
EMPLOYEES table**

MAX(SALARY)
24000

Types of Group Functions

- AVG
- COUNT
- MAX
- MIN
- STDDEV
- SUM



Exercises – Aggregate function

34. Write a SQL Statement that display the category and the count of no of books as 'Book_Count'.

Sort the record on the basis of Book_Count in descending order.

Category	Book_Count
Stories	4
Science	2
Novel	2
Literature	2
Sports	1
Poetry	1
Tales	1

**Select Category, Count(BookCode) as 'Book_Count'
from Book group by Category
order by 2 desc;**

35. Write a SQL Statement that display the category and the count of no of books as 'Book_Count' for every AuthorId. Sort the record on the basis of Book_Count in descending order.

**Select AuthorId, Category, Count(BookCode) as 'Book_Count'
from Book where AuthorId is not null
group by AuthorId, Category
order by 1 asc, 3 desc;**

AuthorId	Category	Book_Count
1	Science	1
2	Stories	3
3	Stories	1
3	Literature	1
3	Poetry	1
4	Literature	1
5	Novel	1
8	Tales	1
9	Science	1

Exercises

36. Write a SQL Statement that display the category and the count of no of books as 'Book_Count' with Book_Count > 1. Sort the record on the basis of Book_Count in descending order.

Category	Book_Count
Stories	4
Science	2
Novel	2
Literature	2

**Select Category, Count(BookCode) as 'Book_Count' from Book
group by Category
having Book_Count > 1
order by 2 desc;**

37. Write a SQL Statement that display the AuthorName, Title of the Book and Category.

Sort the record on the basis of AuthorName in ascending order.

**Select a.AuthorName, b.Title, b.Category
from Author a join Book b
on a.AuthorId = b.AuthorId
order by AuthorName;**

AuthorName	Title	Category
Charles Dickens	Alice in Wonderland	Tales
Dr APJ Kalam	Wings of Fire	Science
Leo Tolstoy	War and Peace	Literature
Mahadevi Verma	Meri Pariwar	Novel
Mary Curie	Radioactivity	Science
R.K. Narayan	Malgudy Days	Stories
R.K. Narayan	Swami and Friends	Stories
R.K. Narayan	The Common Man	Stories
Rabindranath Tagore	Kabuliwala	Stories
Rabindranath Tagore	Geetanjali	Literature
Rabindranath Tagore	The Waterfall	Poetry


Obtaining Data from Multiple Tables

EMPLOYEES

	EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
1	100	King	90
2	101	Kochhar	90
3	102	De Haan	90
...			
18	202	Fay	20
19	205	Higgins	110
20	206	Gietz	110

DEPARTMENTS

	DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID
1	10	Administration	1700
2	20	Marketing	1800
3	50	Shipping	1500
4	60	IT	1400
5	80	Sales	2500
6	90	Executive	1700
7	110	Accounting	1700
8	190	Contracting	1700



	EMPLOYEE_ID	DEPARTMENT_ID	DEPARTMENT_NAME
1	200	10	Administration
2	201	20	Marketing
3	202	20	Marketing
4	124	50	Shipping
5	144	50	Shipping
...			
18	205	110	Accounting
19	206	110	Accounting

Types of Joins

Joins that are compliant with the SQL:1999 standard include the following:

- Natural joins:
 - NATURAL JOIN clause
 - USING clause
 - ON clause
- Outer joins:
 - LEFT OUTER JOIN
 - RIGHT OUTER JOIN
 - FULL OUTER JOIN
- Cross joins

Exercise – Inner Join

38. Write a SQL Statement that display the AuthorName, CountryName and the count of no of books as 'Book_Count'. Sort the record on the basis of Book_Count in descending order.

**Select a.AuthorName, c.CountryName, Count(b.BookCode) as Book_Count
from Author a join Country c
on a.CountryCode = c.CountryCode
join Book b
on a.AuthorId = b.AuthorId
group by a.AuthorName, c.CountryName
order by Book_Count DESC;**

AuthorName	CountryName	Book_Count
R.K. Narayan	India	3
Rabindranath Tagore	India	3
Mary Curie	France	1
Dr APJ Kalam	India	1
Mahadevi Verma	India	1
Leo Tolstoy	Russia	1
Charles Dickens	UK	1

Exercise

39. Write a SQL Statement that display the AuthorName, BirthDate as 'DateOfBirth', Gender and CountryName of those Authors who were born in the months between May and October in any BirthYear. Sort the records on the basis of dates in particular month in ascending order.

```
Select a.AuthorName, Date(a.BirthDate) as DateOfBirth, a.Gender, c.CountryName
from Author a join Country c
on a.CountryCode = c.CountryCode
where Month(a.BirthDate) Between 5 and 10
order by Month(a.BirthDate);
```

AuthorName	DateOfBirth	Gender	CountryName
Rabindranath Tagore	1861-05-07	Male	India
Ernest Hemingway	1899-07-21	Male	USA
Leo Tolstoy	1828-09-09	Male	Russia
R.K. Narayan	1906-10-10	Male	India
Dr APJ Kalam	1931-10-15	Male	India

Exercise – Inner Join, Aggregate Function

40. Write a SQL Statement that display the AuthorName, No of books written as Book_Count, and the price of the expensive book. Sort the records on the basis of Book_Count in ascending order.

```
Select a.AuthorName, Count(b.BookCode) as Book_Count, Max(Price) as Expensive_Book
from Author a join Book b
on a.AuthorId = b.authorId
group by a.AuthorName
order by Book_Count desc;
```

AuthorName	Book_Count	Expensive_Book
R.K. Narayan	3	200.00
Rabindranath Tagore	3	335.00
Mary Curie	1	758.75
Dr APJ Kalam	1	1150.00
Mahadevi Verma	1	235.00
Leo Tolstoy	1	556.50
Charles Dickens	1	100.00

Exercise – Outer Join

41. Write a SQL Statement that display all AuthorName, CountryName and Total books written by him/her as Book_Count. Sort the records on the basis of Book_Count in ascending order.

Select a.AuthorName, c.CountryName, Count(b.BookCode) as Book_Count

from Country c join Author a

on c.CountryCode = a.CountryCode

left outer join Book b

on a.AuthorId = b.AuthorId

group by a.AuthorName, c.CountryName

order by Book_Count DESC;

AuthorName	CountryName	Book_Count
R.K. Narayan	India	3
Rabindranath Tagore	India	3
Mary Curie	France	1
Leo Tolstoy	Russia	1
Mahadevi Verma	India	1
Charles Dickens	UK	1
Dr APJ Kalam	India	1
Lewis Carol	UK	0
Ernest Hemingway	USA	0

Using a Subquery to Solve a Problem

Who has a salary greater than Abel's?

Main query:



Which employees have salaries greater than Abel's salary?

Subquery:



What is Abel's salary?



Subquery

A subquery is a `SELECT` statement that is embedded in the clause of another `SELECT` statement.

The subquery is often referred to as a nested `SELECT`, sub-`SELECT`, or inner `SELECT` statement. The subquery generally executes first, and its output is used to complete the query condition for the main (or outer) query.

You can place the subquery in a number of SQL clauses, including the following:

- `WHERE` clause
- `HAVING` clause
- `FROM` clause

Types of Subqueries

- Single-row subquery



- Multiple-row subquery



Single-Row Subqueries

- Return only one row
- Use single-row comparison operators
=, !=, <, <=, >, >=

Multiple-Row Subqueries

- Return more than one row
- Use multiple-row comparison operators **IN, ANY, ALL**

Exercises - subquery

42. Write a SQL Statement that display the details of the most expensive book.

Select * from Book where Price = (Select Max(Price) from Book)

BookCode	Title	Category	AuthorId	Price	TotalPages
2	Wings of Fire	Science	9	1150.00	2956

43. Write a SQL Statement that display the details of the books written by female author.

Select * from Book where AuthorId IN (Select AuthorId from Author where Gender='Female')

BookCode	Title	Category	AuthorId	Price	TotalPages
1	Radioactivity	Science	1	758.75	1837
3	Meri Pariwar	Novel	5	235.00	316

Exercise – group by, having, subquery, inline view

44. Write a SQL Statement that fetch the details of the authors who have written maximum number of books.

AuthorId	AuthorName	BirthDate	CountryCode	Gender
2	R.K. Narayan	1906-10-10 00:00:00	91	Male
3	Rabindranath Tagore	1861-05-07 00:00:00	91	Male

```
Select * from Author where AuthorId IN
(Select AuthorId from Book where AuthorId is not null
Group by AuthorId
Having Count(BookCode) =(Select Max(Book_Count) from
(Select AuthorId, Count(BookCode) as Book_Count from Book
where AuthorId is not null
Group by AuthorId) T)
);
```

Exercise

45. Write a SQL Statement that create the Player Table including columns PlayerId (primary key), PlayerName, Country, Skill, Runs, Wicket and CaptainId (Foreign Key)

Create Table Player

```
(  
  PlayerId int primary key,  
  PlayerName varchar(30),  
  Country varchar(30),  
  Skill varchar(30),  
  Runs int,  
  Wickets int,  
  CaptainId int references Player(PlayerId)  
);
```

Field	Type	Null	Key
PlayerId	int	NO	PRI
PlayerName	varchar(30)	YES	
Country	varchar(30)	YES	
Skill	varchar(30)	YES	
Runs	int	YES	
Wickets	int	YES	
CaptainId	int	YES	

Exercise

46. Insert records in Player table.

Insert into Player(PlayerId, PlayerName, Country, Skill, Runs, Wickets, CaptainId) values
(1,'Rohit Sharma','India','Batsman', 10709, 9, 1),
(2,'Virat Kohli','India','Batsman',13848, 5, 1),
(3, 'Ravindra Jadega' , 'India', 'AllRounder', 2756, 220, 1),
(4,'Jasprit Bumrah','India','Bowler', 285, 183, 1),
(5,'Eoin Morgan','England','Batsman', 5600, 0, NULL),
(6,'Joe Root','England','Batsman', 7800, 67,5),
(7,'Mark Wood','England','Bowler', 872, 185, 5);

Exercise – self join

47. Write a SQL Statement to fetch the player details consisting of player's name, country, skill, Captain's Name as 'Captain' and Captain's Skill as 'Captain Skill'.

Sort the record in ascending order in the basis of Country Name.

**Select p1.PlayerName, p1.Country, p1.Skill,
p2.PlayerName as 'Captain', p2.Skill as 'Captain Skill'
from Player p1 join Player p2
on p1.CaptainId = p2.PlayerId
order by Country asc;**

PlayerName	Country	Skill	Captain	Captain Skill
Joe Root	England	Batsman	Eoin Morgan	Batsman
Mark Wood	England	Bowler	Eoin Morgan	Batsman
Virat Kohli	India	Batsman	Rohit Sharma	Batsman
Ravindra Jadeja	India	AllRounder	Rohit Sharma	Batsman
Jasprit Bumrah	India	Bowler	Rohit Sharma	Batsman

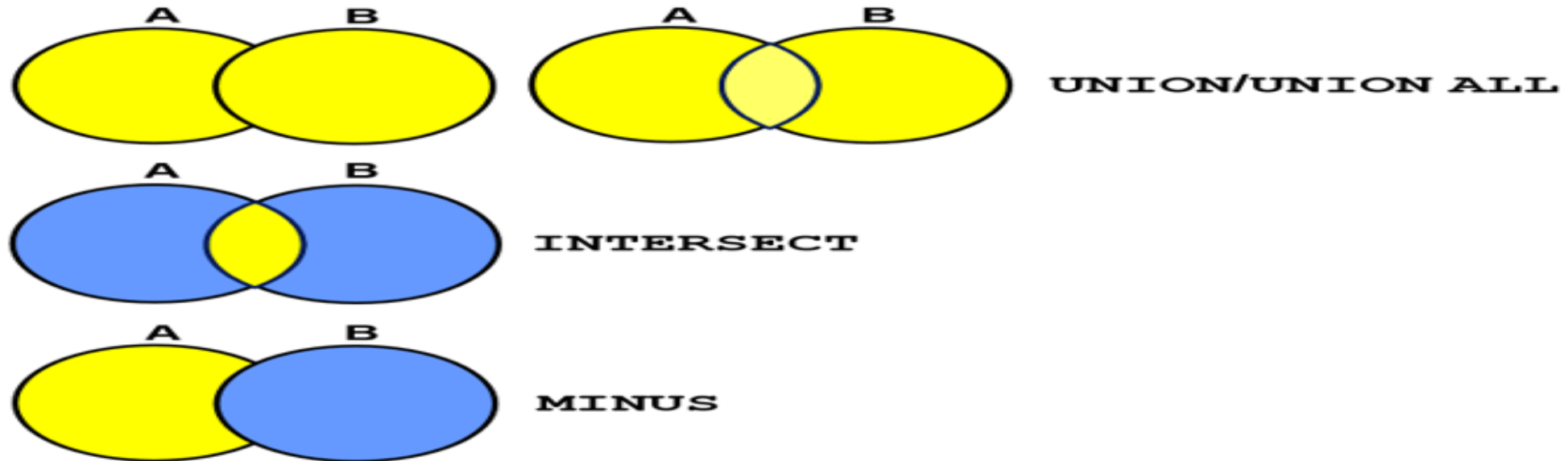
Exercise – self join, case statement

48. Write a SQL Statement to fetch the player details consisting of player name, country, skill, 'Performance' consisting of Runs if the skill is Batsman, Wickets if the skill is Bowler and both Runs and Wickets if the skill is All Rounder and Captain's Name as 'Captain'

```
Select p1.PlayerName, p1.Country, p1.Skill,  
case when p1.Skill = 'Batsman' then concat('Runs -',p1.Runs)  
when p1.Skill = 'Bowler' then concat('Wickets -',p1.Wickets)  
else concat('Runs -',p1.Runs,', Wickets -',p1.Wickets)  
end as 'Performance', p2.PlayerName as 'Captain'  
from Player p1 join Player p2  
on p1.CaptainId = p2.PlayerId;
```

PlayerName	Country	Skill	Performance	Captain
Virat Kohli	India	Batsman	Runs -13848	Rohit Sharma
Ravindra Jadeja	India	AllRounder	Runs -2756, Wickets -220	Rohit Sharma
Jasprit Bumrah	India	Bowler	Wickets -183	Rohit Sharma
Joe Root	England	Batsman	Runs -7800	Eoin Morgan
Mark Wood	England	Bowler	Wickets -185	Eoin Morgan

Set Operators



Set Operators

Set operators combine the results of two or more component queries into one result. Queries containing set operators are called *compound queries*.

Operator	Returns
UNION	Rows from both queries after eliminating duplications
UNION ALL	Rows from both queries, including all duplications
INTERSECT	Rows that are common to both queries
MINUS	Rows in the first query that are not present in the second query

Examples – Union, Union All, Intersect

Create table BookCopy like Book;

Insert into BookCopy Select * from Book;

Delete from BookCopy where BookCode IN (4,5,6);

Insert into BookCopy(BookCode,Title,Category,AuthorId,Price,TotalPages)

VALUES (default,'Isotopes of Radium','Science',1,377.8,583), (default,'Tempest','Tales',null,125,183);

Select * from BookCopy;

BookCode	Title	Category	AuthorId	Price	TotalPages
1	Radioactivity	Science	1	758.75	1837
2	Wings of Fire	Science	9	1150.00	2956
3	Meri Pariwar	Novel	5	235.00	316
7	One More Over	Sports	NULL	38.00	63
8	Kabuliwala	Stories	3	68.00	138
9	Geetanjali	Literature	3	335.00	2238
10	The Waterfall	Poetry	3	25.00	32
11	The Common Man	Stories	2	200.00	275
12	War and Peace	Literature	4	556.50	788
13	Alice in Wonderland	Tales	8	100.00	128
14	Isotopes of Radium	Science	1	377.80	583
15	Tempest	Tales	NULL	125.00	183

Select * from Book;

BookCode	Title	Category	AuthorId	Price	TotalPages
1	Radioactivity	Science	1	758.75	1837
2	Wings of Fire	Science	9	1150.00	2956
3	Meri Pariwar	Novel	5	235.00	316
4	Malgudy Days	Stories	2	156.00	85
5	Swami and Friends	Stories	2	116.00	70
6	Godan	Novel	NULL	160.00	336
7	One More Over	Sports	NULL	38.00	63
8	Kabuliwala	Stories	3	68.00	138
9	Geetanjali	Literature	3	335.00	2238
10	The Waterfall	Poetry	3	25.00	32
11	The Common Man	Stories	2	200.00	275
12	War and Peace	Literature	4	556.50	788
13	Alice in Wonderland	Tales	8	100.00	128

Example – Union, Intersect

Select * from BookCopy

UNION

Select * from Book;

-- select all rows ignoring duplicate rows

BookCode	Title	Category	AuthorId	Price	TotalPages
1	Radioactivity	Science	1	758.75	1837
2	Wings of Fire	Science	9	1150.00	2956
3	Meri Pariwar	Novel	5	235.00	316
7	One More Over	Sports	NULL	38.00	63
8	Kabuliwala	Stories	3	68.00	138
9	Geetanjali	Literature	3	335.00	2238
10	The Waterfall	Poetry	3	25.00	32
11	The Common Man	Stories	2	200.00	275
12	War and Peace	Literature	4	556.50	788
13	Alice in Wonderland	Tales	8	100.00	128
14	Isotopes of Radium	Science	1	377.80	583
15	Tempest	Tales	NULL	125.00	183
4	Malgudy Days	Stories	2	156.00	85
5	Swami and Friends	Stories	2	116.00	70
6	Godan	Novel	NULL	160.00	336

Select * from Book

INTERSECT

Select * from BookCopy;

-- select matched rows only

BookCode	Title	Category	AuthorId	Price	TotalPages
1	Radioactivity	Science	1	758.75	1837
2	Wings of Fire	Science	9	1150.00	2956
3	Meri Pariwar	Novel	5	235.00	316
7	One More Over	Sports	NULL	38.00	63
8	Kabuliwala	Stories	3	68.00	138
9	Geetanjali	Literature	3	335.00	2238
10	The Waterfall	Poetry	3	25.00	32
11	The Common Man	Stories	2	200.00	275
12	War and Peace	Literature	4	556.50	788
13	Alice in Wonderland	Tales	8	100.00	128

Example – Union All

Select * from BookCopy

UNION ALL

Select * from Book;

-- select all rows including duplicate rows

BookCode	Title	Category	AuthorId	Price	TotalPages
1	Radioactivity	Science	1	758.75	1837
2	Wings of Fire	Science	9	1150.00	2956
3	Meri Pariwar	Novel	5	235.00	316
7	One More Over	Sports	NULL	38.00	63
8	Kabuliwala	Stories	3	68.00	138
9	Geetanjali	Literature	3	335.00	2238
10	The Waterfall	Poetry	3	25.00	32
11	The Common Man	Stories	2	200.00	275
12	War and Peace	Literature	4	556.50	788
13	Alice in Wonderland	Tales	8	100.00	128
14	Isotopes of Radium	Science	1	377.80	583
15	Tempest	Tales	NULL	125.00	183
1	Radioactivity	Science	1	758.75	1837
2	Wings of Fire	Science	9	1150.00	2956
3	Meri Pariwar	Novel	5	235.00	316
4	Malgudy Days	Stories	2	156.00	85
5	Swami and Friends	Stories	2	116.00	70
6	Godan	Novel	NULL	160.00	336
7	One More Over	Sports	NULL	38.00	63
8	Kabuliwala	Stories	3	68.00	138
9	Geetanjali	Literature	3	335.00	2238
10	The Waterfall	Poetry	3	25.00	32
11	The Common Man	Stories	2	200.00	275
12	War and Peace	Literature	4	556.50	788
13	Alice in Wonderland	Tales	8	100.00	128

Thanks

