

and the inner query (the correlated subquery) is

```
SELECT AVG(salary)
FROM employees
WHERE department = emp.department
```

- In the above nested query the inner query has to be re-executed for each employee.
- Correlated subqueries may appear elsewhere besides the WHERE clause; for example, this query uses a correlated subquery in the SELECT clause to print the entire list of employees alongside the average salary for each employee's department.
- Again, because the subquery is correlated with a column of the outer query, it must be re-executed for each row of the result.

```
SELECT
employee_number,
name,
(SELECT AVG(salary)
FROM employees
WHERE department = emp.department) AS
department_average
FROM employees AS emp;
```

Syllabus Topic : Views - Concept of View

3.8 Concept of View

➔ (MSBTE - W-13, S-14, S-15, W-15, S-16, W-16)

Q. 3.8.1 What is view? Write the syntax of create view.

(Refer sections 3.8 and 3.8.1)

W-13, S-14, S-15, S-16, W-16, 2 Marks

Q. 3.8.2 Explain views with example.

(Refer section 3.8)

W-15, 4 Marks

DEFINITION

In SQL, a view is a virtual table containing the records of one or more tables based on SQL statement executed.

- Just like a real table, view contains rows and columns. You can add SQL functions, WHERE, and JOIN statements to a view and present the data as if the data were coming from one single table.

- The changes made in a table get automatically reflected in original table and vice versa.

Purpose of View

View is very useful in maintaining the security of database.

Consider a base table employee having following data.

Emp_id	emp_name	Salary	Address
E1	Kunal	8000	Camp
E2	Jay	7000	Tilak Road
E3	Radhika	9000	Somwar Peth
E4	Sagar	7800	Warje
E5	Supriya	6700	LS Road

1. Now just consider we want to give this table to any user but don't want to show him salaries of all the employees. In that we can create view from this table which will contain only the part of base table which we wish to show to the user.

See the following View.

Emp_id	emp_name	Address
E1	Kunal	Camp
E2	Jay	Tilak Road
E3	Radhika	Somwar Peth
E4	Sagar	Warje
E5	Supriya	LS Road

2. Also in multiuser system, it may be possible that more than one user may want to update the data of same table. Consider two users A and B want to update the employee table. In such case we can give views to both these users. These users will make changes in their respective views, and the respective changes are done in the base table automatically.

Syllabus Topic : The Create View Command

3.8.1 The Create View Command

Consider existing table student

Table 3.8.1 : Student

roll_no	stud_name	bdate	marks
101	Kunal	12-02-2000	90
102	Jay	07-08-1999	68
103	Radhika	05-04-2000	85
104	Sagar	13-02-2000	70
105	Supriya	11-08-1999	72

**1. Creating view having all records and fields from existing table****Syntax**

```
CREATE or replace VIEW view_name AS
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

Example

Creating a view of base table student with same structure and all the records.

```
Create or replace view stud_view1
as select * from student;
```

Output**Table 3.8.2 : stud_view1**

roll_no	stud_name	bdate	marks
101	Kunal	12-02-2000	90
102	Jay	07-08-1999	68
103	Radhika	05-04-2000	85
104	Sagar	13-02-2000	70
105	Supriya	11-08-1999	72

2. Creating view having specific fields but all the records from existing table**Syntax**

```
Create or replace view view_name
as select field_1, field_2, ...
from existing_table_name;
```

Example

```
Create or replace view stud_view2
as select roll_no, name from student;
```

Output : The newly created view will be

Table 3.8.3 : stud_view2

roll_no	stud_name
101	Kunal
102	Jay
103	Radhika
104	Sagar
105	Supriya

3. Creating new view having specific records but all the fields from existing table**Syntax**

```
Create or replace view view_name
as select * from existing_table_name
where condition;
```

Example

```
Create or replace view stud_view3
as select * from student
where marks > 80;
```

Output**Table 3.8.4 : stud_view3**

roll_no	stud_name	bdate	marks
101	Kunal	12-02-2000	90
103	Radhika	05-04-2000	85

Q. 3.8.3 Consider following schema.

EMPLOYEE-DETAILS (empname, empId, DOB, salary, job)

Create a view on EMPLOYEE-DETAILS having attribute

(empname, empId, DOB, salary, job) where salary is greater than 20,000

(Refer section 3.8.1)

W-13, W-16, 4 Marks

- Create or replace view v1 as select empname, empId, DOB, salary, job from employee_details where salary > 20,000;

Q. 3.8.4 Consider following schema :

Depositor (Acc_no, Name, PAN, Balance)

Create a view on depositor having attributes (Acc_no, PAN)

where balance is greater than 100000.

(Refer section 3.8.2)

W-14, 4 Marks

Create or replace view v2 as select Acc_no, PAN from depositor where balance > 100000;

Syllabus Topic : Updating Views**3.8.2 Updating Views**

Update query is used to update the records of view. Updation in view reflects the original table also. Means the same changes will be made in the original table also.

Syntax

```
UPDATE view_name
set field_name = new_value;
where condition;
```

Example :

We are updating marks to 73 of student having roll_no 102.

```
UPDATE stud_view1
set marks = 73
where roll_no=102;
```

In this case marks of roll_no 102 will get updated in both view view1 as well as table student.

Output**View - View1**

roll_no	stud_name	bdate	marks
101	Kunal	12-02-2000	90
102	Jay	07-08-1999	73
103	Radhika	05-04-2000	85
104	Sagar	13-02-2000	70
105	Supriya	11-08-1999	72

Output**Table 3.8.5 : Student**

roll_no	stud_name	bdate	Marks
101	Kunal	12-02-2000	90
102	Jay	07-08-1999	73
103	Radhika	05-04-2000	85
104	Sagar	13-02-2000	70
105	Supriya	11-08-1999	72

There are some restrictions on the modification with respect to view.

- In case of view containing joins between multiple tables, only insertion and updation in the view is allowed, deletion is not allowed
- Data modification is not allowed in the view which is based on union queries.
- Data modification is not allowed in the view where GROUP BY or DISTINCT statements are used.
- In view the text and image columns can't be modified.

Syllabus Topic : Views and Joins**3.8.3 Views and Joins**

- We can create views by joining two or more tables.
- Consider following two tables

```
create table employee(
ID number(3),
name varchar (10),
salary number(5) );
create table job(
ID number(5),
title nvarchar (10),
averageSalary number(5))
```

- Now we will create view on these tables

```
CREATE VIEW myView
AS
SELECT e.ID, e.Name, j.title
FROM Employee e , job j
where e.ID = j.ID
```

Syllabus Topic : Views and Sub-queries**3.8.4 Views and Sub-queries**

- We will create view having records of employees working in SMITH's department

Create or replace view v1 as

Select * from emp where deptno =

(select deptno from emp where ename = 'SMITH');

- Once a sequence is created, it is used to return the current value of the sequence.
- To increment the sequence and return the new value, NEXTVAL clause is used.

☞ Syntax

```
CREATE SEQUENCE <Sequence Name>
[INCREMENT BY <Integer Value>
START WITH <Integer Value>
MAXVALUE <Integer Value>
MINVALUE <Integer Value> CYCLE CACHE]
```

☞ Example

```
CREATE SEQUENCE student
MINVALUE 1
MAXVALUE 100
START WITH 1
INCREMENT BY 1
CACHE 20;
```

☞ Increment by

- Increment By clause specifies the time interval between the sequence numbers. Sequence number can be any positive or negative value but not zero. If this clause is omitted, by default value is 1.
- **MAXVALUE and MINVALUE**
This clause is used to specify the maximum or minimum values which are generated by sequences.
- **Start with**
This clause is used to specify the first sequence number generated by using START WITH clause.
- **Cycle** : This clause is used to specify that the sequence continues to generate repeated values.
- **Cache**
This clause is used to specify how many values to generate in advance and those values are kept in memory for faster access of data.

Syllabus Topic : Altering Sequences

3.9.2 Altering Sequences

Q. 3.9.6 How to alter a sequence ?

(Refer section 3.9.2)

(5 Marks)

Sequences can be altered using ALTER SEQUENCE statement. Start value of sequence cannot be altered.

☞ Syntax

```
ALTER SEQUENCE <Sequence Name>
[INCREMENT BY <Integer Value>
MINVALUE <Integer Value>]
```

☞ Example

```
ALTER SEQUENCE student
INCREMENT BY 2
MINVALUE 10
```

☞ Alter sequence

- This statement is used to change the way or action of sequence also used to change increment, minimum or maximum values, and cached numbers of the sequence.
- Alter sequence statement will not affect the past sequence numbers.
- To restart the sequence from the different number you should drop and re-create it again.

Syllabus Topic : Dropping Sequences

3.9.3 Dropping Sequences

Q. 3.9.7 How to drop a sequence?

(Refer section 3.9.3)

(4 Marks)

- DROP SEQUENCE statement is used to remove a sequence from the database
- Restart a sequence by dropping a sequence and then creating a new one.
- E.g. suppose we have a sequence with a current value of 160 and want to restart or re-create a sequence with a value of 30 then you can drop the sequence using DROP SEQUENCE statement and create a new statement with value of 30.

- The sequence must be in your own organization (or structure) for a database. You should have the DROP ANY SEQUENCE in system privilege.

Syntax

- To drop a specific sequence we have to enter following syntax:

```
DROP SEQUENCE sequence_name
```

Example

```
DROP SEQUENCE student;
```

- The above statement drops the sequence supplier_seq.

Syllabus Topic : Indexes - Types, Creating of an Index, Simple, Unique and Composite Index, Dropping Indexes

3.10 Indexes

→ (W-13, S-14, W-14, W-15, S-16, W-16, S-17, W-17)

Q. 3.10.1 Explain index and their types in detail.

(Refer sections 3.10 to 3.10.5)

W-13, W-14, W-15, S-16, W-17, 4 Marks

Q. 3.10.2 Describe following terms :

- Simple Indexes (Refer section 3.10.1)
- Unique indexes (Refer section 3.10.3)
- Composite indexes. (Refer section 3.10.2)

S-14, W-16, S-17, 4 Marks

Q. 3.10.3 Define Index. (Refer section 3.10.1)

S-17, 2 Marks

DEFINITION



Indexes are the special lookup tables which are available to only database search engine for accessing data. Indexes speed up data retrieval effectively.

- An index is a pointer to data in a table. It is similar to the alphabetical index of a book present at the end of book. An index is used to speed up SELECT queries and also WHERE clauses.

- But because of indexes the data input related to INSERT and UPDATE statements get slow down.
- Indexes can be created or dropped with no effect on the data.

Creating Index

CREATE INDEX statement is used to create an index. In this statement we have to mention name of the index, the table and column, and whether the index is in ascending or descending order.

Types of indexes

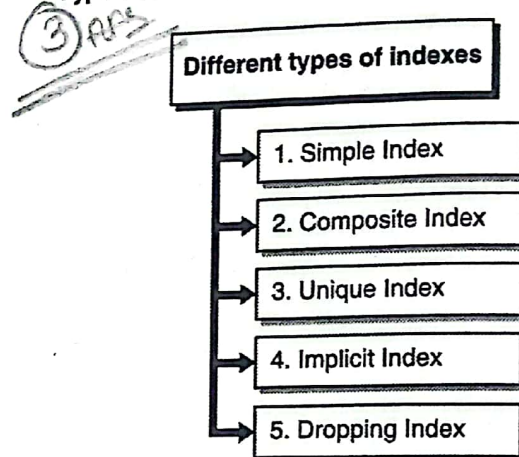


Fig. 3.10.1 : Types of indexes

3.10.1 Simple Index

This is index is created on a single column of a table.

Syntax

```
CREATE INDEX index_name
ON table_name (column_name)
```

Example

```
CREATE INDEX ind1
on student(stud_name);
```

3.10.2 Composite Index

Sometimes duplicate records may available in columns. In such case the composite indexing is better option to index the data. This index is created on a multiple columns of a table.

Syntax

```
CREATE INDEX index_name
ON table_name (column1, column2, ...);
```


Example

```
CREATE INDEX ind2 on student(stud_name , marks);
```

3.10.3 Unique Index

A unique index does not allow any duplicate values to be inserted into the table.

Syntax

```
CREATE UNIQUE INDEX index_name
ON table_name (column1, column2, ...);
```

Example

```
CREATE UNIQUE INDEX ind3
on student(stud_name);
```

3.10.4 Implicit Index

Implicit indexes are indexes that are automatically created by the database server when an object is created. Such indexes are created for primary key and unique constraints

Displaying Index : To display index information regarding table following query is used.

Syntax

```
Show index from table_name;
```

Example

```
Show index from student;
```

3.10.5 Dropping Index**Purpose of Dropping Index**

- To remove an index or domain index from the database you can use the DROP INDEX statement.
- Database invalidates or cancels all objects that depend on the table i.e. Including views, packages, package bodies, functions, and procedures, When you drop an index,.
- The global partitioned index is partitioned by a range or hash partitioned index, if we drop this global partitioned index then all the index partitions also dropped.
- All the index partitions and sub partitions are also dropped, when you drop a composite-partitioned index.

Example

```
DROP INDEX rtt_customer_in_dem;
```

Syllabus Topic : Synonyms - Creating Synonyms, Dropping Synonyms**3.11 Synonyms : Creating Synonyms, Dropping Synonyms**

→ (MSBTE - S-14, W-14, W-17)

Q. 3.11.1 What are synonyms ? Write a syntax for creating a synonym.

(Refer sections 3.11 and 3.11.1)

S-14, 4 Marks

Q. 3.11.2 What are synonyms ? How to create and drop synonym ?

(Refer sections 3.11, 3.11.1 and 3.11.2)

W-14, W-17, 4 Marks

- A synonym is an alias or alternate name for a table, view, sequence, or other schema object.
- They are used mainly to make it easy for users to access database objects owned by other users.
- They hide the underlying object's identity and make it harder for a malicious program or user to target the underlying object.
- Because a synonym is just an alternate name for an object, it requires no storage other than its definition.
- When an application uses a synonym, the DBMS forwards the request to the synonym's underlying base object.
- There are two major uses of synonyms:

Major uses of synonyms

1. Object invisibility

2. Location invisibility

Fig. 3.11.1 : Uses of synonyms

→ 1. Object invisibility

- Synonyms can be created to keep the original object hidden from the user.

→ 2. Location invisibility

- Synonyms can be created as aliases for tables and other objects that are not part of the local database.

3.11.1 Creating Synonyms

- The following code shows how to create a synonym for the employee table:

Syntax

```
CREATE SYNONYM employees for hr.employees;
```

3.11.2 Dropping Synonyms

- The following code shows how drop a synonym:

Syntax

```
DROP SYNONYM employees;
```