Data Control Language

Used to creates privilege to allow users to access and manipulate the database.

Object privileges

- Object privileges vary from object to object
- An owner has all the privileges on the object.
- An owner can give specific privileges on the owner's object

There are two main commands

- GRANT to grant a privilege to the user
- REVOKE to revoke (remove) a privilege from a user

Grant

The GRANT command can be attached to any combination of SELECT, INSERT, UPDATE, DELETE, ALTER

Syntax

- GRANT privilege1,privilege2,privilege3 | ALL ON TABLE|VIEW TO Userid [WITH CHECK OPTION]

Example

- GRANT SELECT ON POLICY TO SAM;
- Grant to ALL Users
  - GRANT SELECT ON POLICY TO PUBLIC;

Revoke

Gets back the given permission from the user

Syntax

- REVOKE privilege1,privilege2,privilege3 | ALL ON table|view FROM userId

Example

- REVOKE SELECT ON POLICY FROM sam;

DATABASE OBJECTS - INTRODUCTION

TEKNOTURF

An Oracle database contains multiple Database objects.

OBJECT	DESCRIPTION
Table	Basic unit of storage; composed of rows and columns
View	Logically represents subsets of data from one or more tables
Sequence	Numeric value generator
Index	Improves the performance of some queries
Synonym	Gives alternative names to objects

View

TEKNOTURF

A view is a virtual table that provides a window through which one can see data (stored in a base relation).

Views contain no data of their own but can be operated on as real relations.

Views can help simplify data access by isolating users from querying details.

Syntax

```
CREATE VIEW viewname AS select query [WITH CHECK OPTION [CONSTRAINT constraint]]  [WITH READ ONLY [CONSTRAINT constraint]];
```

View - Example

TEKNOTURF

Virtual table will point only to the selected records.

PId	PName	PPeriodInYears	MinAmountPerMonth
MBP	Money Back Plan	20	1000
PP	Personal Protect	15	1500

Original table

```
Create view policy_details as select pid,pname from policy;
```

Query to create virtual table

Querying from view

TEKNOTURF

```
Select * from policy_details;
```

PId	PName
MBP	Money Back Plan
PP	Personal Protect

Deny DML operations

TEKNOTURF

To ensure that no DML operations occur through view, create a view with read only option.

```
Create view policy_details as select pid,pname from policy with read only;
```

We cannot insert, update or delete the records from the original table using view.

View Types

TEKNOTURF

Simple view

- Derives data from only one table
- Contains no functions or groups of data
- Can perform DML operations through the view

Complex view

- Derives data from many tables
- Contains functions or groups of data
- Does not always allow DML operations through the view

Complex view - Example

Create view customer_policy_details as Select cname,pname,duedate from customer c join policyenrollment p on c.cid=p.cid join policy pp on p.pid=pp.pid;

Create view policy_details as select pid,pname from policy where MinAmountPerMonth>=500 with check option;

WITH CHECK OPTION is to ensure that all UPDATE and INSERT satisfy the condition(s) in the view definition.

Rules on Views for DML

You can perform DML Operations on a simple view.

You cannot remove a row if the view contains

- Group functions
- GROUP BY clause
- DISTINCT keyword

You cannot modify a row if the view contains

- Group functions
- GROUP BY clause
- DISTINCT keyword
- Column defined by expressions

You cannot add a row if the view contains

- Group functions
- GROUP BY clause
- DISTINCT keyword
- Column defined by expressions
- NOT NULL columns in the base tables not selected in the view.

Inline views

An inline view is a subquery with an alias that can be used within an SQL statement.

A named subquery in the FROM clause of the main query is an example of an inline view.

Removing View

Syntax

- Drop view viewname;

Example

- Drop view policy_details;

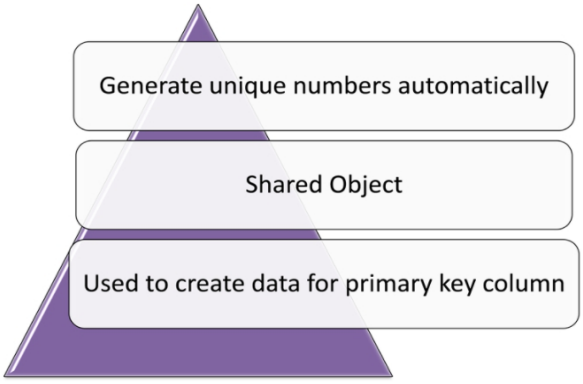
Top N Analysis

Query to display the first three maximum priced policy from the policy table.

```
SELECT ROWNUM as RANK, pname, MinAmontPerMonth
FROM (SELECT pname,MinAmountPerMonth FROM policy
ORDER BY MinAmountPerMonth DESC)WHERE ROWNUM <= 3;
```

Top-n queries retrieve the n largest or smallest values of a column.

Sequence



Sequence

Maxvalue: maximum generated by sequence

Minvalue: minimum sequence value

Cycle : specifies whether to continue generating sequence value after reaching maxvalue

Cache : specifies how many values the Oracle Server pre-allocates and keeps in memory

```
CREATE SEQUENCE seq_name
    [increment by n]
    [start with n]
    [maxvalue n | nomaxvalue]
    [minvalue n | nominvalue]
    [cycle | nocycle]
    [cache | nocache]
```

Sequence Psuedocolumns

NEXTVAL returns the next available sequence value.

- It returns a unique value every time it is referenced, even for different users.

CURRVAL returns the current sequence value.

- NEXTVAL must be issued for that sequence before CURRVAL contains a value.

Sequence - Example

create and use sequence

CREATING A SEQUENCE

```
CREATE SEQUENCE EMP_SEQ START WITH 1;
```

NEXTVAL

```
INSERT INTO EMP VALUES(EMP_SEQ.NEXTVAL,'MINI');
```

CURRVAL

```
SELECT EMP_SEQ.CURRVAL FROM DUAL;
```

Modifying Sequence

TEKNOTURF

Change the increment value, maximum value, minimum value, cycle option, or cache option.

Maxvalue should no be less than the current value

Start with cannot be changed

Future sequence number only will be affected

```
ALTER SEQUENCE emp_seq
INCREMENT BY 20
MAXVALUE 999999
NOCACHE
NOCYCLE;
```

USER_SEQUENCES

TEKNOTURF

How to get the sequence information?

```
select min_value, max_value,
increment_by from user_sequences
where sequence_name='EMP_SEQ ';
```

User_sequences – a data dictionary

How to drop the created sequence?

```
DROP SEQUENCE BOOK_SEQ;
```

Synonym

TEKNOTURF

Tables, views, sequence and other schema object.

Creating an alias Name

What is a synonym?

For what should a synonym be created?

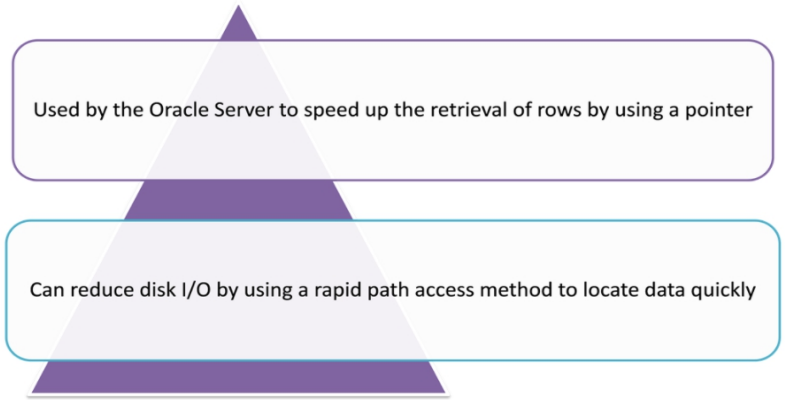
How to create it?

CREATE [PUBLIC] SYNONYM SYN_NAME FOR SCHEMA_OBJ

```
CREATE SYNONYM ENROLL FOR POLICYENROLLMENT;
```

```
DROP SYNONYM ENROLL
```

Index



Index

Create Index

- CREATE [UNIQUE] INDEX INDEX_NAME ON TABLE(COLUMN)
- CREATE INDEX POLICY_IND ON POLICY(PNAME);

You should create an index if:

- A column contains a wide range of values.
- A column contains a large number of null values.
- One or more columns are frequently used together in a WHERE clause or a JOIN condition
- The table is large.

Index Types

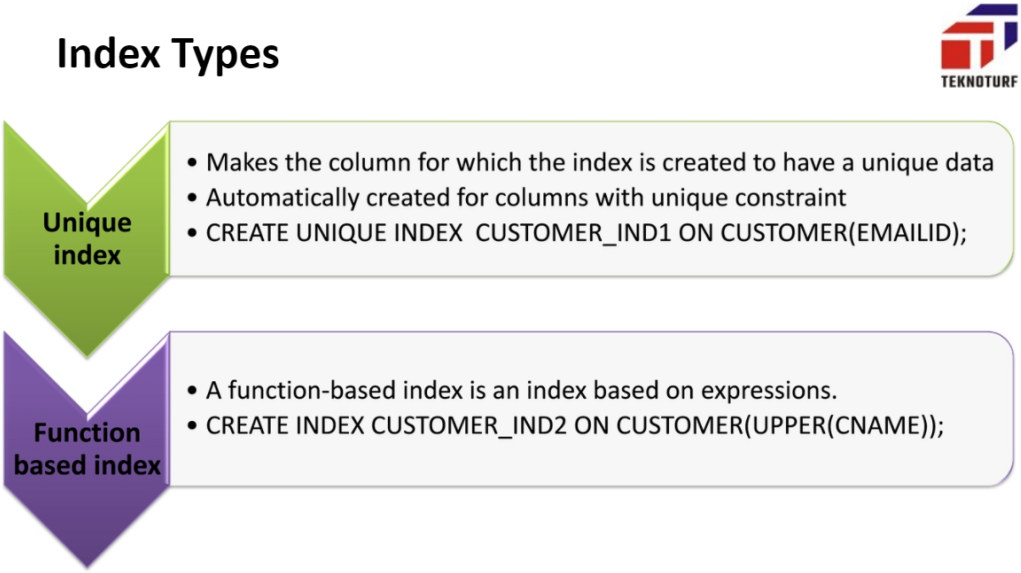
Clustered index

- Creates an index which sorts the data file in the order of the index file
- Can have only one clustered index
- Automatically gets created for Primary key
- CREATE CLUSTERED INDEX POLICY_IND1 ON POLICY(PID);

Non-clustered index

- Can be created for all the columns
- More than 1 index can be created
- Data are not sorted according to the order of index file
- CREATE INDEX POLICY_IND2 ON POLICY(PNAME);

Index Types



Index