**EX.NO.6 Creation of Views, Synonyms, Sequence, Indexes**

**DATE:**

**AIM**:

To execute and verify the SQL commands for Views, Synonyms, Sequence and Indexes.

**PROCEDURE**:

STEP 1: Create the table with its essential attributes.

STEP 2: Insert attribute values into the table.

STEP 3: Create the view from the above created table.

STEP 4: Execute different Commands and extract information from the View.

**COMMANDS**

A SQL View is a virtual table, which is based on SQL SELECT query. Essentially a view is very close to a real database table (it has columns and rows just like a regular table), except for the fact that the real tables store data, while the views don’t. The view’s data is generated dynamically when the view is referenced. A view references one or more existing database tables or other views. In effect every view is a filter of the table data referenced in it and this filter can restrict both the columns and the rows of the referenced tables.

**Create View**: This command is used to create a view.

**CREATE VIEW** view\_name **AS**  
 **SELECT** column\_name(s)  
  **FROM** table\_name WHERE condition;

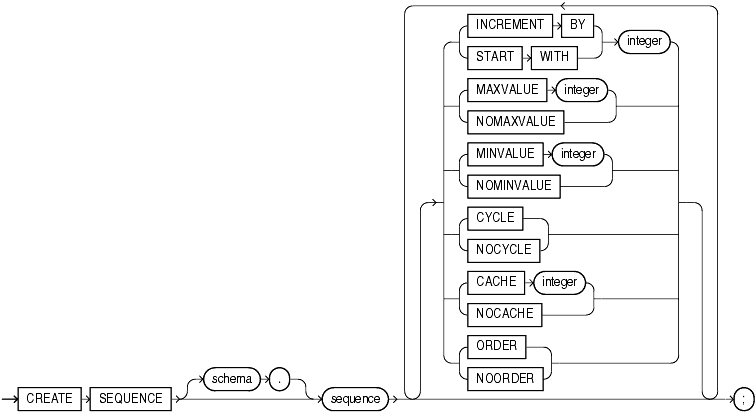
**Update View**: This command is used to update a view.

**CREATE OR REPLACE VIEW** view\_name **AS**  
 **SELECT** column\_name(s)  
  **FROM** table\_name **WHERE** condition;

**Drop View**: This command is used to drop the view table

**DROP VIEW** view\_name;

**SEQUENCE :**



Specify the name of the sequence to be created.

If you specify none of the following clauses, then you create an ascending sequence that starts with 1 and increases by 1 with no upper limit. Specifying only INCREMENT BY -1 creates a descending sequence that starts with -1 and decreases with no lower limit.

* To create a sequence that increments without bound, for ascending sequences, omit the MAXVALUE parameter or specify NOMAXVALUE. For descending sequences, omit the MINVALUE parameter or specify theNOMINVALUE.
* To create a sequence that stops at a predefined limit, for an ascending sequence, specify a value for the MAXVALUE parameter. For a descending sequence, specify a value for the MINVALUE parameter. Also specifyNOCYCLE. Any attempt to generate a sequence number once the sequence has reached its limit results in an error.
* To create a sequence that restarts after reaching a predefined limit, specify values for both the MAXVALUE and MINVALUE parameters. Also specify CYCLE. If you do not specify MINVALUE, then it defaults toNOMINVALUE, which is the value 1.

**For Example**

CREATE SEQUENCE customers\_seq

START WITH 1000

INCREMENT BY 1

NOCACHE

NOCYCLE;

**INDEX**

An index is an optional structure, associated with a table or [table cluster](http://docs.oracle.com/cd/E11882_01/server.112/e25789/glossary.htm#CHDJGGGF), that can sometimes speed data access. By creating an [index](http://docs.oracle.com/cd/E11882_01/server.112/e25789/glossary.htm#i432409) on one or more columns of a table, you gain the ability in some cases to retrieve a small set of randomly distributed rows from the table. Indexes are one of many means of reducing disk I/O.

The following statement creates an index on the  Customer\_Id column of the sample table Orders

CREATE INDEX ord\_customer\_id ON orders (customer\_id);

Multiple indexes can exist for the same table if the permutation of columns differs for each index. You can create multiple indexes using the same columns if you specify distinctly different permutations of the columns. For example, the following SQL statements specify valid permutations:

CREATE INDEX employee\_idx1 ON employees (last\_name, job\_id);

CREATE INDEX employee\_idx2 ON employees (job\_id, last\_name);

# SYNONYM

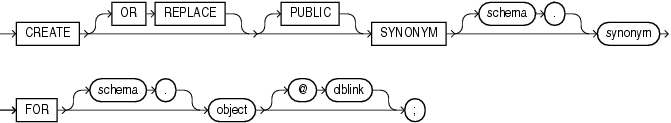
Use the CREATE SYNONYM statement to create a synonym, which is an alternative name for a table, view, sequence, procedure, stored function, package, materialized view, Java class schema object, user-defined object type, or another synonym.

Synonyms provide both data independence and location transparency. Synonyms permit applications to function without modification regardless of which user owns the table or view and regardless of which database holds the table or view. However, synonyms are not a substitute for privileges on database objects. Appropriate privileges must be granted to a user before the user can use the synonym.

To create a private synonym in your own schema, you must have the CREATE SYNONYM system privilege.

To create a private synonym in another user's schema, you must have the CREATE ANY SYNONYM system privilege.

To create a PUBLIC synonym, you must have the CREATE PUBLIC SYNONYM system privilege.



 Examples To define the synonym offices for the table locations in the schema hr, issue the following statement:

CREATE SYNONYM offices From hr. locations;

Table Name hr

Field Name Location

Synonym name offices

Examples To define the synonym offices for the table locations in the schema hr, issue the following statement:

CREATE SYNONYM offices From hr. locations;

Table Name hr

Field Name Location

Synonym name offices

SQL> create table emp\_5(empno number(10),empname varchar2(10),dept varchar2(10),salary number(10));

Table created

SQL> insert into emp\_5 values('&empno','&empname','&dept','&salary');

Enter value for empno: 101

Enter value for empname: ragu

Enter value for dept: it

Enter value for salary: 50000

old 1: insert into emp\_5 values('&empno','&empname','&dept','&salary')

new 1: insert into emp\_5 values('101','ragu','it','50000')

1 row created.

SQL> insert into emp\_5 values('&empno','&empname','&dept','&salary');

Enter value for empno: 102

Enter value for empname: ram

Enter value for dept: cse

Enter value for salary: 60000

old 1: insert into emp\_5 values('&empno','&empname','&dept','&salary')

new 1: insert into emp\_5 values('102','ram','cse','60000')

1 row created.

SQL> insert into emp\_5 values('&empno','&empname','&dept','&salary');

Enter value for empno: 103

Enter value for empname: ragavan

Enter value for dept: ece

Enter value for salary: 40000

old 1: insert into emp\_5 values('&empno','&empname','&dept','&salary')

new 1: insert into emp\_5 values('103','ragavan','ece','40000')

1 row created.

SQL> select \* from emp\_5;

EMPNO EMPNAME DEPT SALARY

---------- ---------- ---------- ----------

101 ragu it 50000

102 ram cse 60000

103 ragavan ece 40000

VIEW:

SQL> create view rrrr as select \* from emp\_5 where dept='cse';

View created.

SQL> update rrrr set empname='rathika' where empno=102;

1 row updated.

SQL> select \* from emp\_5;

EMPNO EMPNAME DEPT SALARY

---------- ---------- ---------- ----------

101 ragu it 50000

102 rathika cse 60000

103 ragavan ece 70000

SQL> drop view rrrr;

View dropped.

INDEXES:

SQL> create index pindex on emp\_5(empno,empname,dept,salary);

Index created.

SQL> drop index pindex;

Index dropped.

SEQUENCES:

SQL> create sequence emp\_seq minvalue 1 start with 1 increment by 1 cache 20;

Sequence created.

SQL> insert into emp\_5(empno,empname,dept,salary)values(emp\_seq.nextval,'yokesh','eee',30000);

1 row created.

SQL> select \*from emp\_5;

EMPNO EMPNAME DEPT SALARY

---------- ---------- ---------- ----------

101 ragu it 50000

102 ram cse 60000

103 ragavan ece 40000

1 rathika eee 30000

2 yokesh eee 30000

SQL> drop sequence emp\_seq;

Sequence dropped.

SYNONYM:

SQL> create synonym emp\_5 for employee;

Synonym created.