Database Systems Laboratory 8

View, Sequence, Synonym & Index

Chittaranjan Pradhan
School of Computer Engineering,
KIIT University

View, Sequence, Synonym & Index

View

View

View is an object which gives the user a logical view of data from an underlying table or tables

You can restrict what users can view by allowing them to see only a few columns from a table

When a view is created from more than one table, the user can view data from the view without using join conditions and complex conditions

Views also hide the names of the underlying tables

View is stored as a SELECT statement in the Data Dictionary View contains no data of its own Any updation of rows in the table will automatically reflect in the views

A query fired on a view will run slower than a query fired on a base table

View...

View Types

Views are of two types:

- ***Simple view***: It is based on one table. It allows data manipulation
- ***Complex view***: It is based on one or more tables. It doesn't allow data manipulation

View Creation

The syntax for creating a view is:
CREATE [OR REPLACE] VIEW viewname AS SELECT statement;

*CREATE VIEW Debts AS SELECT * FROM BILLS;*

View...

Creating View from another View

Similar to the view creation from tables, a view can be created from a previously created view

*CREATE VIEW Credit_dbt AS SELECT * FROM Debts WHERE account_id=4;*

Displaying the content of a view

User can display the content from a view as:
SELECT * FROM viewname;

*SELECT * FROM Debts;*

Inserting into table through view

User can insert records into the underlying table through the view as:
INSERT INTO viewname VALUES(val1, val2..);

INSERT INTO Debts VALUES(val1,val2,..);

View...

Describing a view structure

View structure can be described as:
DESCRIBE viewname;

DESCRIBE Debts;

Updating table data through view

User can update the records of a table through the view as:
UPDATE viewname SET columnname=newvalue[WHERE *cond*^{n}**];**

*UPDATE Debts SET PRICE=PRICE*1.1 WHERE PUBLISHER='MGH';*

Deleting records from table through view

User can delete records from table through view as:
DELETE FROM viewname [WHERE *cond*^{n}**];**

DELETE FROM Debts WHERE condn;

View...

Creating a View WITH CHECK OPTION Constraint

This constraint applies to the WHERE clause condition in the subquery. It allows insertion and updation of rows based on the condition that satisfies the view

*CREATE VIEW Asd AS SELECT * FROM Products WHERE price < 15;*

*CREATE VIEW Psd AS SELECT * FROM Products WHERE price < 15 WITH CHECK OPTION CONSTRAINT con1;*

Creating a View WITH READ ONLY Constraint

This option is used to make sure that the data in the underlying table are not changed through the view

*CREATE VIEW Cheap_products_view3 AS SELECT * FROM Products WHERE price < 15 WITH READ ONLY CONSTRAINT cheap_products_view3_read_only;*

View...

Viewing all the user views

All user created views can be displayed as:
SELECT * FROM USER_VIEWS;

Removing a View

A view can be removed as:
DROP VIEW viewname;

DROP VIEW Debts;

View...

Altering a View

When the underlying table is altered, the view becomes invalid. Thus, the view requires the recompilation as:
ALTER VIEW viewname COMPILE;

ALTER VIEW Debts COMPILE;

ALTER VIEW statement lets you add or remove constraints to or from a view

ALTER VIEW Psd DROP CONSTRAINT con1;

View...

Use of GROUP BY clause

GROUP By clause can be used with view creation

CREATE OR REPLACE VIEW Vn (empno, noincr, amount) AS SELECT emp_no, COUNT(*), SUM(amt) FROM INCR GROUP BY emp_no;

CREATE VIEW Pr AS SELECT product_type_id, AVG(price) average_price FROM Products WHERE price < 15 GROUP BY product_type_id HAVING AVG(price) > 13;

Complex View

Complex View

It is based on one or more tables. It doesn’t allow data manipulation

In complex view, you can use only SELECT statement

CREATE VIEW Vw AS SELECT P.name, PT.type,P.price FROM PRODUCTS P NATURAL JOIN PRODUCTTYPE PT;

Advantages of Views

Advantages of Views

Some of the major advantages of using views are:

- Views allow in setting up different security levels for the same base table, thus protecting certain data from people who do not have proper authority
- The views allow the same data to be seen by different users in different ways at the same time
- Views can be used to hide complex queries

Sequence

Sequence

Sequence is used to generate a sequence of numbers. The value generated can have a maximum of 38 digits

The minimum information required to generate numbers using a sequence are:

- The starting number
- The maximum number
- The increment value

The syntax for creating a sequence is:
CREATE SEQUENCE seqname INCREMENT BY n START WITH s MAXVALUE m1 \ NOMAXVALUE MINVALUE m2 \ NOMINVALUE [CYCLE \ NOCYCLE] [CACHE c \ NOCACHE];

Sequence...

CREATE SEQUENCE sq INCREMENT BY 1 START WITH 100 MAXVALUE 999 NOCACHE;

CURRVAL & NEXTVAL pseudocolumns

NEXTVAL column returns the next available number in the sequence

CURRVAL column gives the current sequence value

NEXTVAL must be used at least once to get the value from CURRVAL

SELECT sq.CURRVAL FROM DUAL;
SELECT sq.NEXTVAL FROM DUAL;
INSERT INTO emp(eid) VALUES (sq.NEXTVAL);

Sequence...

Viewing the details of a user sequences

SELECT sequence_name, last_number, max_value, min_value, increment_by FROM USER_SEQUENCES;

Modifying a Sequence

Modification of a sequence does not allow you to change the START WITH option. Similarly, the maximum value cannot be set to a number less than the current number
ALTER SEQUENCE seqname INCREMENT BY n MAXVALUE m1 \ NOMAXVALUE MINVALUE m2 \ NOMINVALUE [CYCLE \ NOCYCLE] [CACHE c \ NOCACHE];

Dropping a Sequence

A sequence can be dropped as:
DROP SEQUENCE sequencename;

DROP SEQUENCE sq;

Synonym

Synonym

Synonyms are used to create alternate names for tables, views, sequences ... etc. The syntax for this is:
CREATE [PUBLIC] SYNONYM synname FOR objectname;

CREATE SYNONYM Emp FOR Employee;

CREATE SYNONYM Cstd FOR Customer_Details;
*SELECT * FROM Cstd;*

Viewing the details of a user synonyms

SELECT synonym_name, table_name, table_owner FROM USER_SYNONYMS;

Dropping a Synonym

A Synonym can be dropped as:
DROP SYNONYM synonymname;

DROP SYNONYM Emp;

Index

Index is used for faster retrieval of rows from a table. It can be used implicitly or explicitly. Mainly, index is of two types:

Simple Index

It is created on a single column. The syntax is:
CREATE INDEX indexname ON tablename(column);

CREATE INDEX idx ON Student (cgpa);

Complex Index

It is created on more than one column. The syntax is:
CREATE INDEX indexname ON tablename(columns);

CREATE INDEX ids ON Student (first, last);

Index...

Viewing the details of a user-defined index

SELECT index_name, table_name FROM USER_INDEXES;

SELECT index_name, table_name FROM USER_INDEXES WHERE table_name= 'Student';

Rebuilding an Index

When a table goes through changes, it is advisable to rebuild indexes based on that table. The syntax is:
ALTER INDEX indexname REBUILD;

ALTER INDEX ids REBUILD;