

Introduction

- VIEW select query stored in the database
 serves as virtual table based on results of a real SQL select
 - view is not part of physical schema of database
- contains columns and rows, but values are from one or more "real" tables of database
- always contains up to date (actual) values (data)
 - DBMS for all queries on view runs stored selects, therefore gets actual data

Differences between view and table

- data in table can be directly modified using DML statements of SQL (possibilities of modifications data on view are limited)
- view does not take up almost any space on disk
- obtaining results from more complicated (and joined) views may be more time consuming than with direct access to a table
 o performance can be increased by usage of proper indices

Benefits of using the view

- can represent a subset of data from a particular table
- can easily join multiple tables into one virtual table
- can be used as aggregation of tables, where there can be calculated columns
- appears as a simple table, although it may contain complicated query (select)
- occupies very little space
- provides more security (read-only access for most of types)

Creation of new view

 syntax for view creation CREATE VIEW name of view AS SELECT columns FROM table WHERE conditions • example CREATE VIEW nmb of subjects AS SELECT name, surname, (SELECT COUNT (*) FROM subject attendance WHERE id student = id) AS number FROM student

WITH CHECK OPTION

flag for the CREATE VIEW statement
for inserted/updated value, the WHERE condition clause must apply
example:

CREATE VIEW old_users AS SELECT * FROM FBUser
WHERE birthday < '01.01.88'
WITH CHECK OPTION;
does not allow for example:
insert into old_users
values ('milanko', 'Milan', 'Nosal', '01.02.88', 'M');

The view can be defined on another view

• in FROM clause can be used name of another view

• reasons for this approach:

- complexity control complex query can be replaced by a series of queries built on one another
- coordinate two parts of the app change in one part is automatically reflected in another part
- beware of loops! interdependence of two views

views as layers (1st layer - base 'real' table)

Usage of view

- can be used in same way as name of any other table in select query
 - SELECT * FROM name of view
- príklad
 - SELECT * FROM nmb of subjects;

Removing of view

- cascade removal if the base table/view is removed, all views that depend on it are removed (not in Oracle)
- syntax for removal of view
 - DROP VIEW name;
- príklad
 - DROP VIEW nmb_of_subjects;

Modification of values in view

 view can be modified if SELECT does not use DISTINCT, aggregate functions, set operators, ORDER BY, GROUP BY, FROM cannot use multiple tables, WHERE cannot have subqueries and all NOT NULL columns are part of view o for exceptions (e.g., update over multiple Oracle tables), see the documentation • views can be made modifiable by use of triggers

Materialized view

- a database object containing data from provided select query at the time of creation ("basic" view gets data only after usage of view)
 - if query is computationally demanding, it can be preferable to have data prepared for frequent queries
- materialized view must be refreshed

 e.g. automatically after commit to tables
 over which view is defined; at given time
 intervals; manually

