

Ministerul Educației al Republicii Moldova
Universitatea Tehnică a Moldovei
Facultatea de Calculatoare, Informatică și Microelectronică
Filiera Anglofonă "Computer Science"

Views in SQL

Course Thesis

Baze de Date și Cunoștințe

Student: Vlas Mihai, FAF-141

Conducător: Irina Cojanu

Chișinău 2017

Contents

Introduction	4
1 Views in SQL	5
2 Views in Oracle 12c	9
Conclusions	10
References	11

Introduction

SQL stands for Structured Query Language, is a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system. Originally based upon relational algebra and tuple relational calculus, SQL consists of a data definition language, data manipulation language, and data control language. The scope of SQL includes data insert, query, update and delete, schema creation and modification, and data access control. Although SQL is often described as, and to a great extent is, a declarative language (4GL), it also includes procedural elements.

SQL was one of the first commercial languages for Edgar F. Codd's relational model, as described in his influential 1970 paper, "A Relational Model of Data for Large Shared Data Banks." Despite not entirely adhering to the relational model as described by Codd, it became the most widely used database language.

SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987. Since then, the standard has been revised to include a larger set of features. Despite the existence of such standards, most SQL code is not completely portable among different database systems without adjustments. The SQL language is subdivided into several language elements, including:

Clauses, which are constituent components of statements and queries. (In some cases, these are optional.) Expressions, which can produce either scalar values, or tables consisting of columns and rows of data. Predicates, which specify conditions that can be evaluated to SQL three-valued logic (3VL) (true/false/unknown) or Boolean truth values and are used to limit the effects of statements and queries, or to change program flow. Queries, which retrieve the data based on specific criteria. This is an important element of SQL. Statements, which may have a persistent effect on schemata and data, or may control transactions, program flow, connections, sessions, or diagnostics. SQL statements also include the semicolon (";") statement terminator. Though not required on every platform, it is defined as a standard part of the SQL grammar. Insignificant whitespace is generally ignored in SQL statements and queries, making it easier to format SQL code for readability.

1 Views in SQL

In SQL, a view is a virtual table based on the result-set of an SQL statement. A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database. 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. A view can contain all rows of a table or select rows from a table. A view can be created from one or many tables which depends on the written SQL query to create a view.

Views, which are a type of virtual tables allow users to do the following:

- Structure data in a way that users or classes of users find natural or intuitive.
- Restrict access to the data in such a way that a user can see and (sometimes) modify exactly what they need and no more.
- Summarize data from various tables which can be used to generate reports.

Besides the standard role of basic user-defined views, SQL Server provides the following types of views that serve special purposes in a database:

- **Indexed Views:** An indexed view is a view that has been materialized. This means the view definition has been computed and the resulting data stored just like a table. You index a view by creating a unique clustered index on it. Indexed views can dramatically improve the performance of some types of queries. Indexed views work best for queries that aggregate many rows. They are not well-suited for underlying data sets that are frequently updated.
- **Partitioned Views:** A partitioned view joins horizontally partitioned data from a set of member tables across one or more servers. This makes the data appear as if from one table. A view that joins member tables on the same instance of SQL Server is a local partitioned view.
- **System Views:** System views expose catalog metadata. You can use system views to return information about the instance of SQL Server or the objects defined in the instance. For example, you can query the sys.databases catalog view to return information about the user-defined databases available in the instance.

Table 1.1 – Customers

ID	NAME	AGE	ADDRESS	SALARY
1	ION	25	Chisinau	3000
2	Victor	30	Orhei	2000
3	Mihai	20	Balti	4000

Table 1.2 – CustomersView

NAME	AGE
ION	25
Victor	30
Mihai	20

There are several actions that we can do with views in SQL:

- 1) **Create View**
- 2) **Update View**
- 3) **Delete View**

First of all let's analyze how the views are created and what kind of syntax we need for it. Database views are created using the CREATE VIEW statement. Views can be created from a single table, multiple tables or another view.

To create a view, a user must have the appropriate system privilege according to the specific implementation.

```

1 CREATE VIEW view_name AS
2 SELECT column1, column2, ...
3 FROM table_name
4 WHERE condition;
```

Listing 1.1 – Views creating statements [2]

Let's take an example of creating a view from a table:

Now let's create the view that will have only Customer's name and age:

```

1 SQL > CREATE VIEW CUSTOMERS_VIEW AS
2 SELECT name, age
3 FROM CUSTOMERS;
```

Listing 1.2 – Creating view from a table [2]

The result will be the following:

Table 1.3– UpdatedView

NAME	AGE
ION	25
Victor	35
Mihai	20

Next option for Views let's analyze the updating action.

First of all to update a view we need to accomplish several conditions, which are:

- The SELECT clause may not contain the keyword DISTINCT.
- The SELECT clause may not contain summary functions.
- The SELECT clause may not contain set functions.
- The SELECT clause may not contain set operators.
- The SELECT clause may not contain an ORDER BY clause.
- The FROM clause may not contain multiple tables.
- The WHERE clause may not contain subqueries.
- The query may not contain GROUP BY or HAVING.
- Calculated columns may not be updated.
- All NOT NULL columns from the base table must be included in the view in order for the INSERT query to function.

So, if a view satisfies all the above-mentioned rules then it can be updated. The following listing has an example to update the age of Victor.

```
1 SQL > UPDATE CUSTOMERS_VIEW
2     SET AGE = 35
3     WHERE name = 'Victor';
```

Listing 1.3– Updating a View [2]

The last operation is Deleting a View, which we can perform by dropping it if it's no longer needed.

```
1 DROP VIEW view_name;
```

Listing 1.4– Drop View [2]

For our example, the sequence will be:

```
1 DROP VIEW CUSTOMERS_VIEW;
```

Listing 1.5– Drop CustomerView [2]

Table 1.4– UpdatedView

NAME	AGE
ION	25
Victor	35

Treating Views as simple tables allows us to insert new rows and delete some specific rows or columns from the view. This can be done with INSERT/DELETE command.

```
1 SQL > DELETE FROM CUSTOMERS_VIEW
2 WHERE age = 20;
```

Listing 1.6– Drop CustomerView [2]

Note!: For INSERT Command, are applied the same rules as for UPDATE command.
Let's delete a row which has recorded the age=20.

2 Views in Oracle 12c

Conclusions

Views in SQL are a powerful tool that can simplify the work of the database architect or developer. By combining multiple tables in one view, easily updating and deleting some rows is facilitating the work and makes it a lot easier than using typical update and insert commands directly in the table of work. Views are statements with a name associated to it and stored in the database, this allows an easy creation and drop of the view in any moment of work. Depending on the type of view we need to use, the specific attributes should be modified in views.

References

- 1 Aldebran Robotics, *official page*, www.aldebaran.com/en
- 2 Aldebran Robotics, *NAO Software 2.1.3 documentation* www.doc.aldebaran.com/2-1/index.html
- 3 OpenCV Dev Team, *OpenCV 2.4.9.0 documentation*, www.opencv.org/modules/refman.html
- 4 Timo Ojala, *Multiresolution gray-scale and rotation invariant texture classification with local binary patterns*, 2002
- 5 Jan Erik Solem, *Programming Computer Vision with Python*, 2012
- 6 M. A. Turk and A. P. Pentland, *Face Recognition Using Eigenfaces*, 1991.
- 7 S. Suhas, A. Kurhe, Dr.P. Khanale, *Face Recognition Using Principal Component Analysis and Linear Discriminant Analysis on Holistic Approach in Facial Images Database*, Vol. 2, December 2012
- 8 W. Zhao, R. Chellappa, P. J. Phillips and A. Rosenfeld, *Face recognitions literature survey*, ACM Computing Surveys, Vol. 35, No. 4, December 2003, pp. 399–458
- 9 Sukhan Lee, Hyungsuck Cho, Kwang-Joon Yoon, Jangmyung Lee, *Intelligent Autonomous Systems 12*, Volume 1
- 10 Shah Alam, Malaysia Shamsuddin, *Face detection technique of Humanoid Robot NAO for application in robotic assistive therapy*, Center of Excellence for Humanoid Robot Bio-Sensing (HuRoBs), Univ. Teknol. MARA
- 11 Xavier Alameda-Pineda, Vasil Khalidov, Radu Horaud and Florence Forbes, *Finding Audio-Visual Events in Informal Social Gatherings*
- 12 Jan Cech, Ravi Mittal, Antoine Deleforge, *Active-Speaker Detection and Localization with Microphones and Cameras Embedded into a Robotic Head*, Humanoids 2013 - IEEE-RAS International Conference on Humanoid Robots, Oct 2013, Atlanta, United States. 2013
- 13 Xavier Alameda-Pineda, Radu Horaud, *Vision-Guided Robot Hearing*, The International Journal of Robotics Research, SAGE Publications (UK and US), 2015