



HY360 ΑΡΧΕΙΑ ΚΑΙ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ
ΦΡΟΝΤΙΣΤΗΡΙΟ ΣΤΗ ΓΛΩΣΣΑ SQL
VIEWS

ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ
ΥΠΟΛΟΓΙΣΤΩΝ

OUTLINE

➤ SQL VIEWS

- SQL CREATE VIEW Statement
- SQL UPDATE A VIEW
- SQL DROP A VIEW

➤ SQL MATERIALIZED VIEWS

- CREATE MATERIALIZED VIEW

SQL VIEWS

Τι είναι τα VIEWS:

- Ένα VIEW είναι ένας εικονικός πίνακας βασισμένος στο αποτέλεσμα ενός SQL Statement
- Ένα VIEW περιέχει στήλες και γραμμές όπως ένα κανονικό table
- Τα πεδία ενός VIEW είναι αποτέλεσμα ενός ή και περισσότερων πινάκων από τη βάση δεδομένων

SQL VIEWS

Χρησιμότητα των VIEWS:

- Δομούν τα δεδομένα με τρόπο κατανοητό για τους χρήστες
- Προσφέρουν ασφάλεια με το να περιορίζουν τη πρόσβαση στα δεδομένα με τρόπο ώστε ένας χρήστης να μπορεί να δει ή κάποιες φορές και να κάνει μετατροπή ακριβώς στα δεδομένα που χρειάζεται και σε τίποτα παραπάνω
- Συλλέγουν δεδομένα από διαφορετικούς πίνακες κρύβοντας την πολυπλοκότητά τους από τον χρήστη

SQL VIEWS

CREATE A VIEW:

```
CREATE VIEW view_name AS  
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

Example:

```
CREATE VIEW DetailsView AS  
SELECT NAME, ADDRESS  
FROM StudentDetails  
WHERE S_ID < 4;
```

StudentDetails Table

S_ID	NAME	ADDRESS
1	Pavlos	Satha
2	Roula	Satha
3	Georgia	62 Martiron
4	Kostas	Therissou

StudentMarks Table

ID	NAME	MARKS	AGE
1	Paulos	70	19
2	Roula	70	20
3	Goergia	80	19
4	Dimitris	90	21

SQL VIEWS

RESULT:

```
SELECT * FROM DetailsView;
```

Result Table

NAME	ADDRESS
Pavlos	Satha
Roula	Satha
Georgia	62 Martiron

SQL VIEWS

CREATE A VIEW FROM MULTIPLE TABLES Example:

```
CREATE VIEW MarksView AS
SELECT StudentDetails.NAME, StudentDetails.ADDRESS, StudentMarks.MARKS
FROM StudentDetails, StudentMarks
WHERE StudentDetails.NAME = StudentMarks.NAME;
```

Result

S_ID	NAME	ADDRESS
1	Pavlos	Satha
2	Roula	Satha
3	Georgia	62 Martiron

SQL VIEWS

UPDATE A VIEW:

```
CREATE OR REPLACE VIEW view_name AS  
SELECT column1,column2,..  
FROM table_name  
WHERE condition;
```

StudentDetails Table

S_ID	NAME	ADDRESS
1	Pavlos	Satha
2	Roula	Satha
3	Georgia	62 Martiron
4	Kostas	Therissou

StudentMarks Table

ID	NAME	MARKS	AGE
1	Paulos	70	19
2	Roula	70	20
3	Goergia	80	19
4	Dimitris	90	21

SQL VIEWS

UPDATE A VIEW Example:

```
CREATE OR REPLACE VIEW MarksView AS
```

```
SELECT StudentDetails.NAME, StudentDetails.ADDRESS, StudentMarks.MARKS,  
StudentMarks.AGE
```

```
FROM StudentDetails, StudentMarks
```

```
WHERE StudentDetails.NAME = StudentMarks.NAME;
```

Result:

```
SELECT * FROM MarksView;
```

Result Table

ID	NAME	MARKS	AGE
1	Paulos	70	19
2	Roula	70	20
3	Goergia	80	19

SQL VIEWS

DROP A VIEW:

DROP VIEW `view_name`;

Example

DROP VIEW `MarksView`;

SQL MATERIALIZED VIEWS

Τι είναι ένα Materialized View:

- Ένα Materialized View είναι ένα View που αποθηκεύει το αποτέλεσμα ενός View query.
- Όταν κάνουμε επερώτηση σε ένα Materialized View μας επιστρέφει τα δεδομένα μαζί με το ίδιο το View.
- Ένα Materialized View μοιάζει περισσότερο με ένα κανονικό Table.

SQL MATERIALIZED VIEWS

Πλεονεκτήματα χρήσης Materialized View:

- Βελτιώνει την απόδοση ενός Query
- Απλοποιεί ένα Query

Μειονεκτήματα χρήσης Materialized View:

- Χρειάζεται Manual Update στα δεδομένα που περιέχει
- Πιθανή ύπαρξη ασυμφωνίας δεδομένων μεταξύ του Materialized View και των δεδομένων των Tables από το οποίο έχει δημιουργηθεί

SQL MATERIALIZED VIEWS

Διαφορές μεταξύ Materialized View και ενός View:

Component	View	Materialized View
Stored on database or disk	SQL query definition	SQL query definition, and data that is returned by it
Data storage	None – remains in underlying tables	Stored within the view
Refresh	Latest data is returned whenever the query is run	Data stored in view may not be the most up-to-date. Options can be set to update the materialized view data periodically.
Performance	Same as underlying query	Usually faster than a view.
Indexes	None – uses same indexes as underlying tables	Can be created on the materialized view itself
Space Used	Small, as it is only the SQL query definition	Can be large, as the data is stored with the object

SQL MATERIALIZED VIEWS

Παράδειγμα χρήσης **Materialized View**:

```
CREATE MATERIALIZED VIEW empdept AS
SELECT e.emp_id, e.last_name, e.salary, d.dept_id, d.dept_name
FROM employee e
INNER JOIN department d ON e.dept_id = d.dept_id;
```

emp_id	last_name	dept_id	salary
1	Jones	3	23000
2	Smith	2	49000
3	King	1	21000

Employee Table

dept_id	dept_name
1	Sales
2	Customer Service
3	Finance

Department Table

SQL MATERIALIZED VIEWS

Result:

```
SELECT emp_id, last_name, salary, dept_id, dept_name  
FROM empdept;
```

Result Table

emp_id	last_name	salary	dept_id	dept_name
1	Jones	23000	3	Finance
2	Smith	49000	2	Customer Service
3	King	21000	1	Sales

SQL MATERIALIZED VIEWS

Thank you !