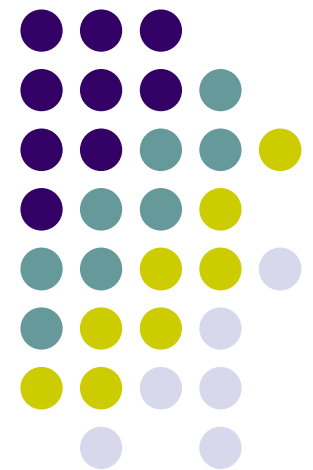


Views

CIS 331: Introduction to Database Systems





Topics:

- Views
- COLUMN ... FORMAT ...
- Oracle system views:
 - USER_TABLES
 - USER_VIEWS
 - USER_OBJECTS



Views

- Let us find the list of students who are not taking CIS525:

```
SELECT ssn
  FROM students
  WHERE ssn NOT IN
    (
      SELECT student
        FROM students_classes
        WHERE class = 'CIS525'
    )
;
```



Creating views

- We can save the result of this query in a *view*, so we can access it any time:

```
CREATE VIEW v1
AS
  SELECT ssn
    FROM students
      WHERE ssn NOT IN
        (
          SELECT student
            FROM students_classes
              WHERE class = 'CIS525'
        )
;
```



Creating views

- Or, alternatively, we can use this piece of code to create a view:

```
CREATE VIEW v2  
AS  
    SELECT ssn  
        FROM students  
MINUS  
    SELECT student  
        FROM students_classes  
        WHERE class = 'CIS525'  
;
```



Views

- Views v1 and v2 seem to be equivalent in terms of content, as can be verified by:

```
SELECT *  
    FROM v1;  
SELECT *  
    FROM v2;
```

- Are they equivalent in terms of what you can do with them?



Views

- Let us perform some basic operations on them:

```
UPDATE v1
  SET ssn = '1111111111'
  WHERE ssn = '1259875398';
SELECT *
  FROM v1;
```

```
UPDATE v2
  SET ssn = '2222222222'
  WHERE ssn = '3459875398';
SELECT *
  FROM v2;
```



Views

- Because the way we have defined v2 involves operations on 2 tables, we are not allowed to manipulate it.
- Oracle avoids ambiguity with respect to which of the 2 tables participating in the MINUS it should update.
- Notice that even though we were not able to update v2 the ssn in v2 has changed. Why? Check the underlying table:

```
SELECT ssn  
FROM students;
```



USER_TABLES view

- Let us take a look at the *user_tables* view:

```
SELECT *  
FROM user_tables;
```

- This was probably too much information at one time.

```
DESC user_tables;
```

```
SELECT table_name, tablespace_name  
FROM user_tables  
;
```

- Why is USER_TABLES a view and not a table?



USER_VIEWS view

- Similarly:

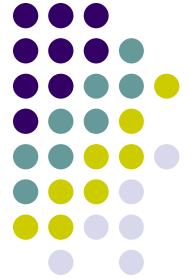
```
DESC user_views;
```

```
SET LONG 1000
```

```
SELECT view_name, text  
       FROM user_views
```

```
;
```

```
SET LONG 80
```



USER_OBJECTS view

- Similarly:

```
SELECT object_name, object_type
       FROM user_objects
;
```

- This is wrapping around the screen and is difficult to read. Let us make it more legible and repeat the query:

```
COLUMN object_name FORMAT A30;
```

```
SELECT object_name, object_type
       FROM user_objects
;
```



TAB view

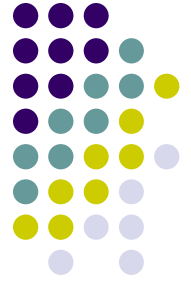
- Remember TAB, which we used to see all the tables we have in our account?

```
SELECT *  
FROM tab;
```

- TAB is just a view:

```
SELECT object_name, object_type  
FROM user_objects  
WHERE object_type IN ('TABLE', 'VIEW')  
  
;
```

Dropping views



- And finally...

```
DROP VIEW v1;
```

```
DROP VIEW v2;
```