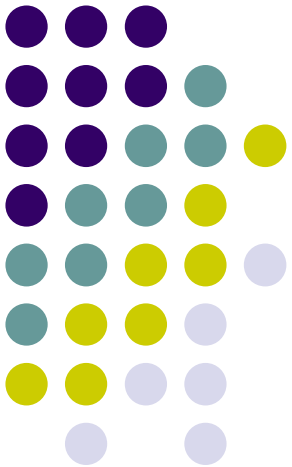
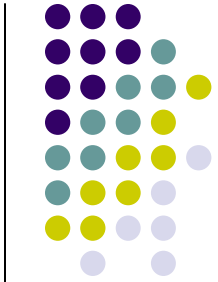
Indexes

CIS 331: Introduction to Database Systems

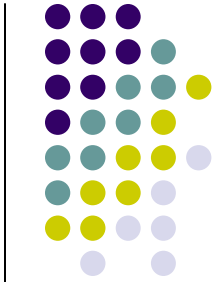


Topics:



- Indexes (or indices, plural of index) in Oracle
- Create, alter, drop
- Function-based indexes
- USER_INDEXES
- Indexes and primary keys

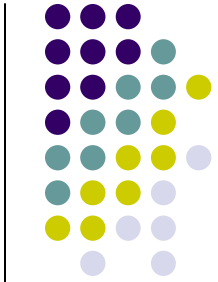
Create index



- You can create an index that references any combination of attributes on any table, as long as there are no other indexes that reference the same combination of attributes on the same table.

```
CREATE INDEX prof_names1
    ON professors (first_name, last_name)
;
```

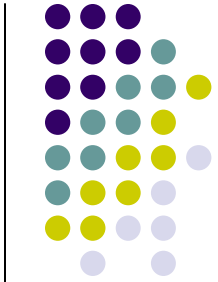
Create index



- The same attributes as before, on a different table, are perfectly fine:

```
CREATE INDEX stud_names1
    ON students (first_name, last_name)
;
```

Create index

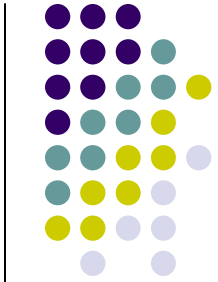


- Order matters. So, this is different from:

```
CREATE INDEX stud_names2
    ON students (last_name, first_name)
;
```

- Hence, you cannot create an index that references only one column in a table if another index already references only that column in that table. (Why?)

Create index



- Uniqueness

```
CREATE UNIQUE INDEX stud_names1_uniq
    ON students (first_name, last_name)
;
```

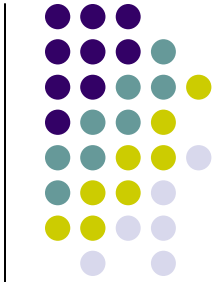
- Ooooups…

```
DROP INDEX stud_names1;

CREATE UNIQUE INDEX stud_names1_unik
    ON students (first_name, last_name)
;
```

- You should not use indexes to enforce uniqueness of your data. Use constraints instead.

Alter index

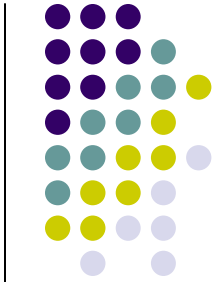


- Alter index allows you to modify indexes.

```
ALTER INDEX stud_names1_unik
            RENAME TO stud_names1_uniq
;
```

- Alter index however allows more sophisticated modifications... Examples an be found in Oracle SQL Reference document, under "ALTER INDEX"

Function-based indexes



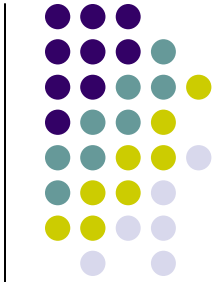
- Function-based indexes (not sure if I have sufficient privileges to do that, but let me try...)

```
CREATE INDEX stud_upper_name
    ON students (UPPER(last_name))
;
```

- This index would speed up a query such as:

```
SELECT *
    FROM students
        WHERE UPPER(last_name) like 'D%'
;
```

Function-based indexes



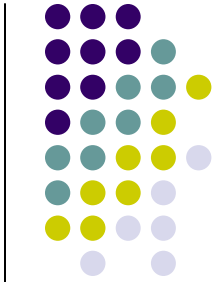
- Function-based indexes can become very creative. Unfortunatelly, our simple Students DB does not allow this level of creativity:

```
CREATE INDEX idx
     ON table_1 (a + b * (c - 1), a)
 ;
```

- which would speed up a query of the sort:

```
SELECT a FROM table_1
     WHERE a + b * (c - 1) < 100
 ;
```

USER_INDEXES



- Remember Oracle system views, such as USER_TABLES, USER_VIEWS and USER_OBJECTS?

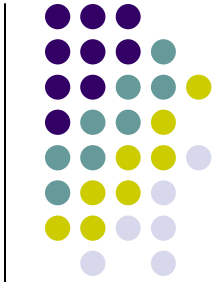
  As you might have guessed, you can find out more about your indexes in the USER_INDEXES view.

```
DESC user_indexes;

COLUMN index_name FORMAT A20;
COLUMN index_type FORMAT A8;
COLUMN table_name FORMAT A20;

SELECT index_name, index_type, table_name, uniqueness
    FROM user_indexes
;
```

Indexes and PK's

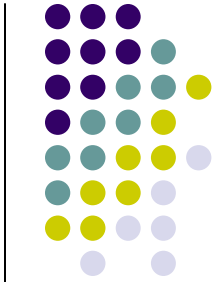


- What you can conclude from this listing? What is the relationship between indexes and PK's ?

- So, if we tried to create an index which references a PK:

```
CREATE INDEX stud_ssns
    ON students (ssn)
;
```

Dropping an index



- And finally:

```
DROP INDEX prof_names1;
DROP INDEX stud_names1_uniq;
DROP INDEX stud_names2;
DROP INDEX stud_upper_name;
```