Views

Sample database

```
CREATE TABLE Department(
name                VARCHAR2(50),
code                CHAR(5),
total_staff_number NUMBER(2)                NOT NULL,
Chair         VARCHAR2(50),
budget              NUMBER(9,1)             NULL,
CONSTRAINT dept_pkey PRIMARY KEY(name),
CONSTRAINT dept_ckey1 UNIQUE(code),
CONSTRAINT dept_ckey2 UNIQUE(chair),
CONSTRAINT dept_check1
CHECK (total_staff_number BETWEEN 1 AND 50)  );
```

```
CREATE TABLE Course(
c#                  CHAR(7),
title               VARCHAR2(200)           NOT NULL,
credits             NUMBER(1)               NOT NULL,
offered_by          VARCHAR2(50)            NULL,
CONSTRAINT course_pkey PRIMARY KEY(c#),
CONSTRAINT course_check1
        CHECK (credits IN (6, 12) ),
CONSTRAINT course_fkey1 FOREIGN KEY(offered_by)
        REFERENCES Department(name) );
```


View ? What is it and why do we need it ?

View is a logical "window" on selected data from a relational database

We need views to separate database users from implementation details, to enhance access control, to simplify queries

Views versus relational tables:

CREATE TABLE ⇒ real table

CREATE VIEW ⇒ virtual table

Example

Find the names of all departments together with the total number of courses offered by each department

Department

name | code | total staff number | chair | budget

Course

c# | title | credits | offered by

```
CREATE VIEW VDCNT( name, total_courses ) AS
      ( SELECT name,count(c#)
        FROM Department LEFT OUTER JOIN Course
        ON Department.name = Course.offered_by
        GROUP BY name );
```


Example

Find the names of all departments together with the total number of courses offered by each department

```
CREATE VIEW VDCNT( name, total_courses ) AS
     ( SELECT name,count(title)
       FROM Department LEFT OUTER JOIN Course
       ON Department.name = Course.offered_by
       GROUP BY name );
```

VDCNT

name | total_courses

Keywords

Find the names of all departments together with the total number of courses offered by each department

```
CREATE VIEW VDCNT( name, total_courses ) AS
     ( SELECT name,count(title)
       FROM Department LEFT OUTER JOIN Course
       ON Department.name = Course.offered_by
       GROUP BY name );
```

View name

Find the names of all departments together with the total number of courses offered by each department

```
CREATE VIEW VDCNT( name, total_courses ) AS
     ( SELECT name,count(title)
       FROM Department LEFT OUTER JOIN Course
       ON Department.name = Course.offered_by
       GROUP BY name );
```


View schema

Find the names of all departments together with the total number of courses offered by each department

```
CREATE VIEW VDCNT( name, total_courses ) AS
     ( SELECT name,count(title)
       FROM Department LEFT OUTER JOIN Course
       ON Department.name = Course.offered_by
       GROUP BY name );
```


View definition (query)

Find the names of all departments together with the total number of courses offered by each department

```
CREATE VIEW VDCNT( name, total_courses ) AS
     ( SELECT name,count(title)
       FROM Department LEFT OUTER JOIN Course
       ON Department.name = Course.offered_by
       GROUP BY name );
```


Application of a view

Find the names of all departments together with the total number of courses offered by each department

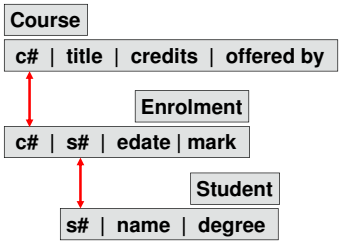
```
CREATE VIEW VDCNT( name, total_courses ) AS
     ( SELECT name,count(title)
       FROM Department LEFT OUTER JOIN Course
       ON Department.name = Course.offered_by
       GROUP BY name );
```

Find a department that offers more than 5 courses

```
SELECT name
FROM VDCNT
WHERE total_courses > 5;
```


Views as "procedures"

Find the names of all students who collected more credit points than a student with number 879546



Views as "procedures"

Find the names of all students who collected more credit points than a student with number 879546

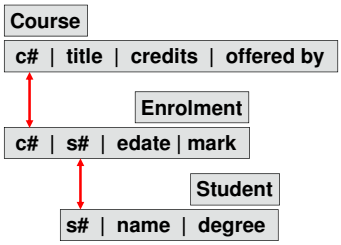
Q1: Find the total number of credit points collected by each student

Q2: Use Q1 to find the total number of credit points collected by a student with number 879546

Q3: Use Q1 and Q2 to find all students who collected more credit points than a student with number 879546

Views as "procedures"

Q1: Find the total number of credit points collected by each student



Views as "procedures"

Q1: Find the total number of credit points collected by each student

Q11: Construct a view that consists of two columns: s# and credits obtained for all completed courses

Q12: Group Q11 on s# and perform summation of all credit points earned by each student

Views as "procedures"

Q11: Construct a view that consists of two columns: s# and credits obtained for all completed courses

Course | c# | title | credits | offered by

Enrolment | c# | s# | edate | mark

```
CREATE VIEW Scredits( s#, credits ) AS
     ( SELECT Enrolment.s#, credits
       FROM Enrolment JOIN Course
       ON Enrolment.c# = Course.c#
       WHERE mark IS NOT NULL AND mark > 49);
```

Scredits | s# | credits

Views as "procedures"

Q12: Group Q11 on s# and perform summation of all credit points earned by each student

Scredits | s# | credits

```
CREATE VIEW Totcredits( s#, total_credits) AS
     ( SELECT s#, SUM(credits)
       FROM Scredits
       GROUP BY s# );
```

Totcredits | s# | total_credits

Views as "procedures"

Find the names of all students who collected more credit points than a student with number 879546

Q1: Find the total number of credit points collected by each student DONE!

Q2: Use Q1 to find the total number of credit points collected by a student with number 879546

Q3: Use Q1 and Q2 to find all students who collected more credit points than a student with number 879546

Views as "procedures"

Q2: Use Totcredits to find the total number of credit points collected by a student with number 879546

Totcredits | s# | total_credits

```
CREATE VIEW V879546(total_credits) AS
     ( SELECT total_credits
       FROM Totcredits
       WHERE s# = 879546);
```

V879546 | total_credits

Views as "procedures"

Find the names of all students who collected more credit points than a student with number 879546

Q1: Find the total number of credit points collected by each student

DONE !

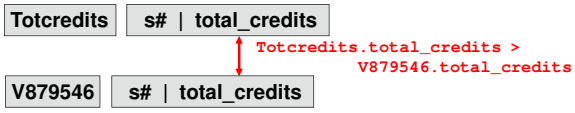
Q2: Use Q1 to find the total number of credit points collected by a student with number 879546

DONE !

Q3: Use Q1 and Q2 to find all students who collected more credit points than a student with number 879546

Views as "procedures"

Q3: Use Totcredits and V879546 to find all students who collected more credit points than a student with number 879546




```
SELECT Totcredits.s#
FROM Totcredits JOIN V879546
ON Totcredits.total_credits >
                    V879546.total_credits;
```


View update problem

Is it possible to update the contents of a database through a view ?

Usually NOT, sometimes YES !

It is possible to update the contents of a database through a view if a view is defined on only one relational table, it does not contain computable attributes, and its schema includes all attributes from the relational table it is defined on

View update problem

```
CREATE VIEW Totcredits( s#, total_credits) AS
     ( SELECT s#, sum(credits)
       FROM Scredits
       GROUP BY s# );
```

```
UPDATE Totcredits
SET total_credits = total_credits + 6;
```

Who suppose to get extra 6 credits ?

References

Elmasri R., Navathe S. B., *Database Systems*, chapter 5.3

Ramakrishnan R., Gehrke J., *Database Management Systems*, chapter 3.6,

https://sai.uow.edu.au/oradocs/
SQL Reference, CREATE VIEW statement