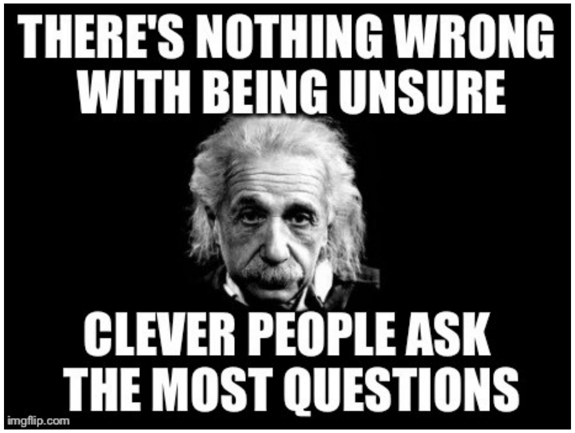
SQL Query **Part 1**

Asking Questions in Class

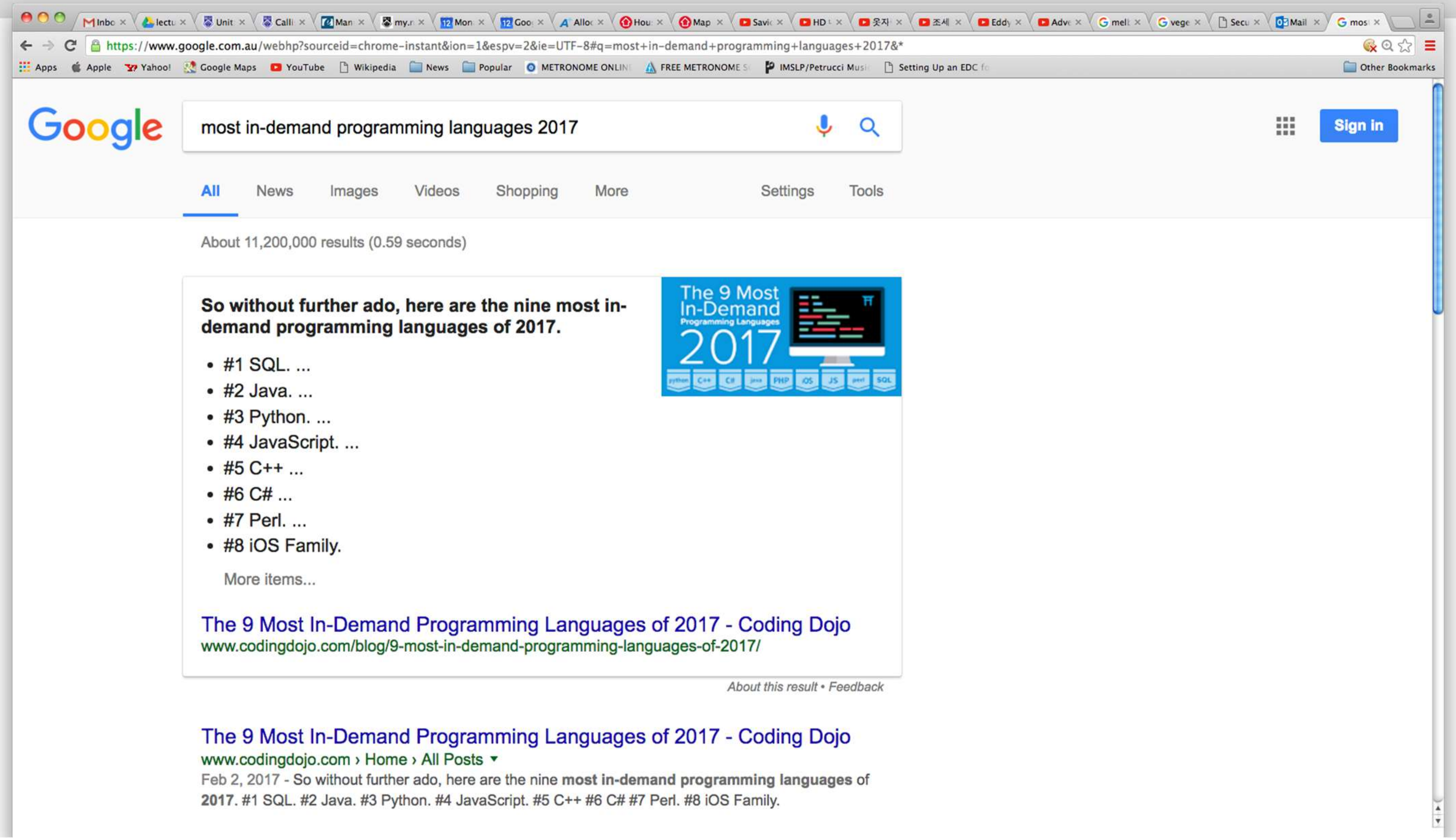
- **Traditional way (preferred)**
  - Say it!
- **Ask online**
  - Go to the link shown on Google slide
  - Write your questions
    - Explain your question clearly
  - And/or upvote/downvote already posted questions
  - You can post question anonymously
  - Do not misuse the system: We can look at the logs to identify users if required


THERE'S NOTHING WRONG WITH BEING UNSURE
CLEVER PEOPLE ASK THE MOST QUESTIONS
imgflip.com

Kahoot!

- We will play a round of Kahoot every week
- Questions will be based on the content covered during the lecture
- Top-3 will get chocolates

or simply “well done” if I forget to bring chocolates :P



most in-demand programming languages 2017

About 11,200,000 results (0.59 seconds)

So without further ado, here are the nine most in-demand programming languages of 2017.

- #1 SQL. ...
- #2 Java. ...
- #3 Python. ...
- #4 JavaScript. ...
- #5 C++ ...
- #6 C# ...
- #7 Perl. ...
- #8 iOS Family.

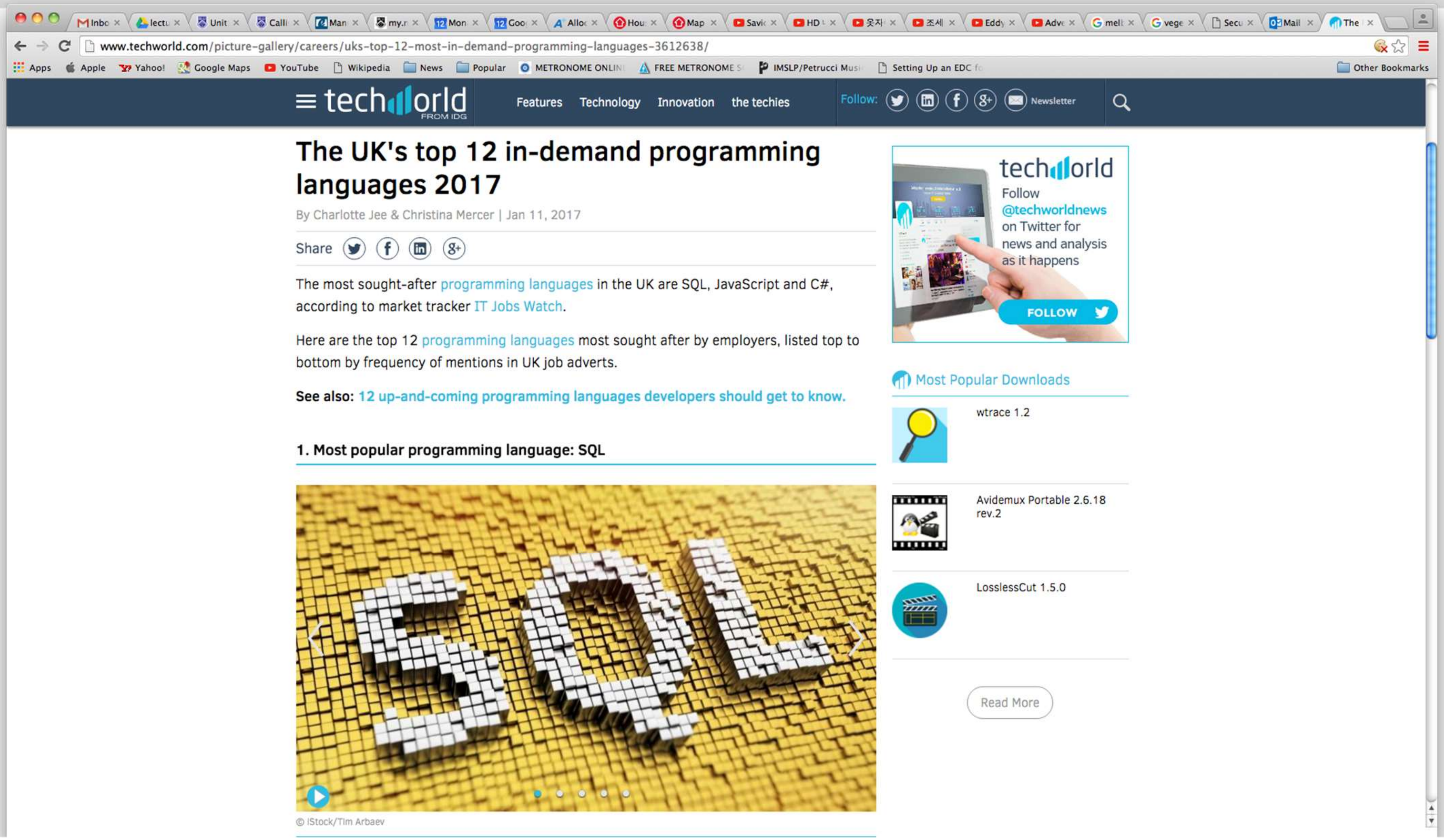
More items...

The 9 Most In-Demand Programming Languages of 2017 - Coding Dojo
www.codingdojo.com/blog/9-most-in-demand-programming-languages-of-2017/

About this result • Feedback

The 9 Most In-Demand Programming Languages of 2017 - Coding Dojo
www.codingdojo.com › Home › All Posts ▾
Feb 2, 2017 - So without further ado, here are the nine **most in-demand programming languages** of **2017**. #1 SQL. #2 Java. #3 Python. #4 JavaScript. #5 C++ #6 C# #7 Perl. #8 iOS Family.

Screen shot 2017-03-12 at 10.09.25 PM



The UK's top 12 in-demand programming languages 2017

By Charlotte Jee & Christina Mercer | Jan 11, 2017

Share

The most sought-after programming languages in the UK are SQL, JavaScript and C#, according to market tracker IT Jobs Watch.

Here are the top 12 programming languages most sought after by employers, listed top to bottom by frequency of mentions in UK job adverts.

See also: 12 up-and-coming programming languages developers should get to know.

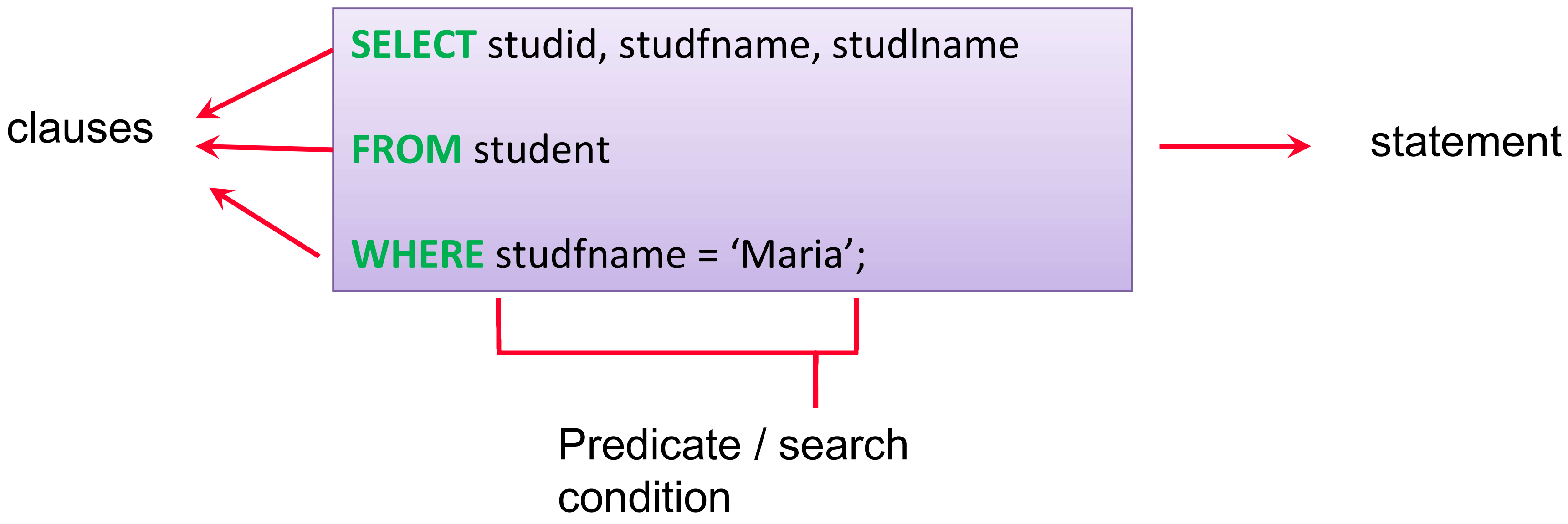
1. Most popular programming language: SQL

© iStock/Tim Arbaev

techworld
Follow @techworldnews on Twitter for news and analysis as it happens
FOLLOW
Most Popular Downloads
wtrace 1.2
Avidemux Portable 2.6.18 rev.2
LosslessCut 1.5.0
Read More

Screen shot 2017-03-12 at 10.12.57 PM

Anatomy of an SQL SELECT Statement



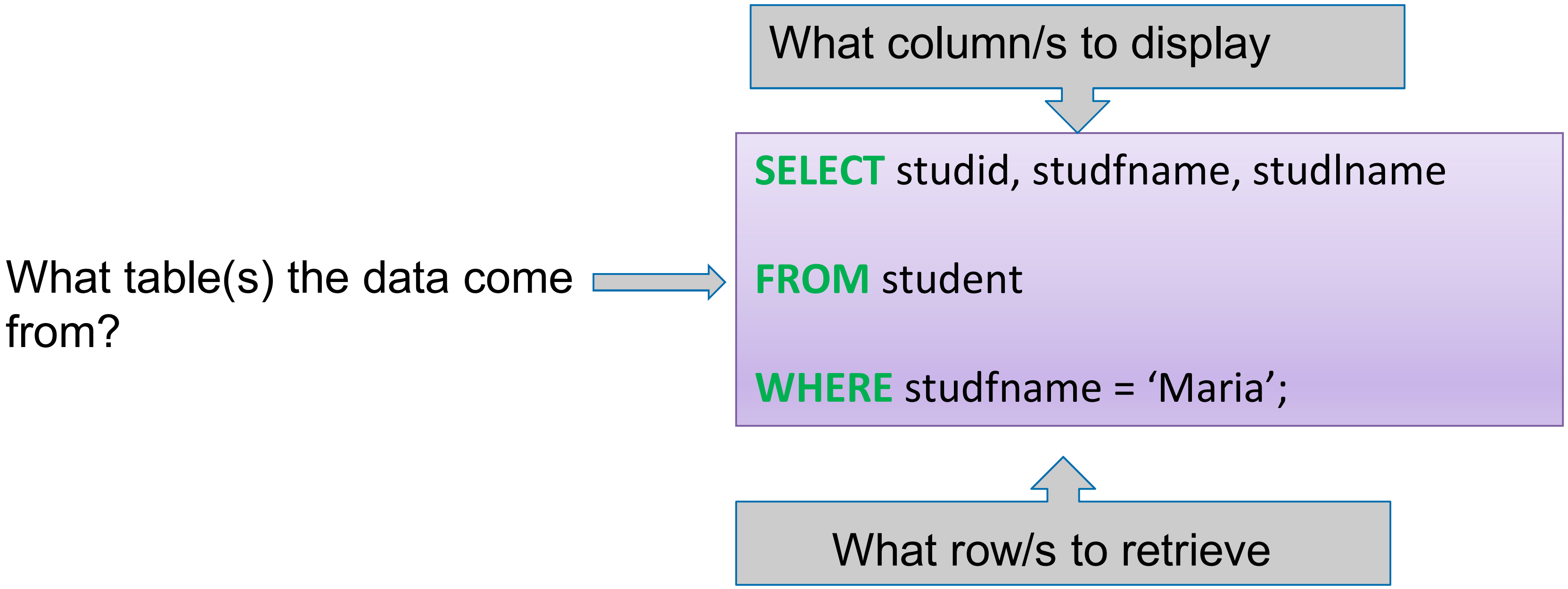
```
SELECT studid, studfname, studlname

FROM student

WHERE studfname = ‘Maria’;
```

clauses

statement

Predicate / search condition

SQL SELECT Statement - Usage



What column/s to display

```
SELECT studid, studfname, studlname

FROM student

WHERE studfname = ‘Maria’;
```

What table(s) the data come from?

What row/s to retrieve

SQL Predicates or Search Conditions

- The search conditions are applied on **each** row, and the row is returned if the search conditions are evaluated to be TRUE for that row.
- **Comparison**
  - Compare the value of one expression to the value of another expression.
  - Operators:
    - =, < >, <, >, !=, <=, >=
  - Example: salary > 5000
- **Range**
  - Test whether the value of an expression falls within a specified range of values.
  - Operators:
    - BETWEEN
  - Example: salary BETWEEN 1000 AND 3000 (both are inclusive)

SQL Predicates or Search Conditions

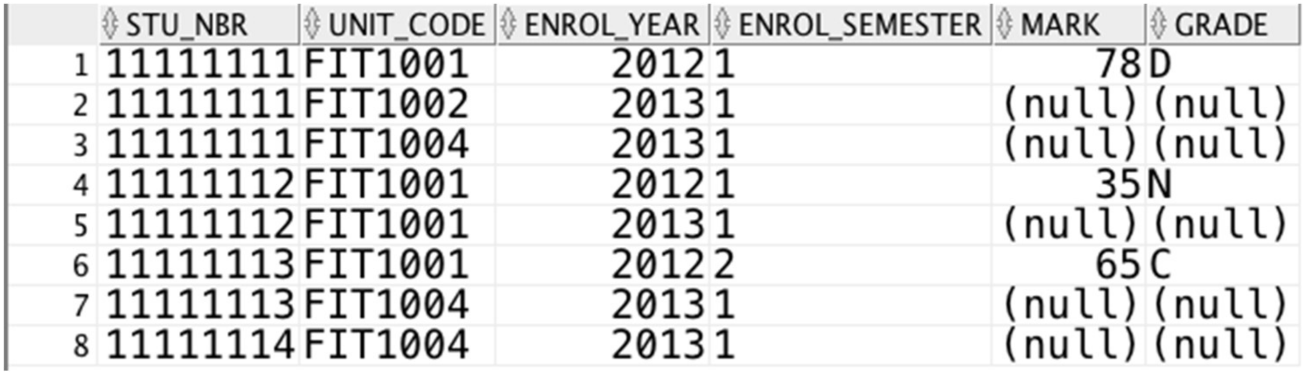
- **Set Membership**
  - To test whether the value of expression equals one of a set of values.
  - Operator:
    - IN
  - Example : city IN (‘Melbourne’, ’Sydney’)
- **Pattern Match**
  - To test whether a string (text) matches a specified pattern.
  - Operator:
    - LIKE
  - Patterns:
    - % character represents **any sequence of zero or more** character.
    - _ character represents **any single** character.
  - Example:
    - WHERE city LIKE ‘M%’
    - WHERE unitcode LIKE ‘FIT91__’

SQL Predicates or Search Conditions

- NULL
  - To test whether a column has a NULL (unknown) value.
  - Example: WHERE grade IS NULL.
- Use in subquery (to be discussed in the future)
  - ANY, ALL
  - EXISTS

What row will be retrieved?

- Predicate evaluation is done using three-valued logic. TRUE, FALSE and UNKNOWN
- DBMS will evaluate the predicate against each row.
- Row that is evaluated to be TRUE will be retrieved.
- NULL is considered to be UNKNOWN.



	STU_NBR	UNIT_CODE	ENROL_YEAR	ENROL_SEMESTER	MARK	GRADE
1	11111111	FIT1001	2012	1	78	D
2	11111111	FIT1002	2013	1	(null)	(null)
3	11111111	FIT1004	2013	1	(null)	(null)
4	11111112	FIT1001	2012	1	35	N
5	11111112	FIT1001	2013	1	(null)	(null)
6	11111113	FIT1001	2012	2	65	C
7	11111113	FIT1004	2013	1	(null)	(null)
8	11111114	FIT1004	2013	1	(null)	(null)

2. Consider the predicate “mark >= 50”, what row(s) will be selected for this predicate by DBMS?

a. 1, 4 and 6
b. All rows
c. 1 and 6
d. All rows except row 4

Recall that SQL returns only the rows for which the predicate is evaluated to be TRUE

Combining Predicates

- Logical operators
  - AND, OR, NOT
- Rules:
  - An expression is evaluated LEFT to RIGHT.
  - Sub-expression in brackets are evaluated first.
  - NOTs are evaluated before AND and OR
  - ANDs are evaluated before OR.

Truth Table

- AND is evaluated to be TRUE if and only if **both** conditions are TRUE
- OR is evaluated to be TRUE if and only if at least one of the conditions is TRUE

AND

B A	T	U	F
T	T	U	F
U	U	U	F
F	F	F	F

OR

B A	T	U	F
T	T	T	T
U	T	U	U
F	T	U	F

T = TRUE
F = FALSE
U = Unknown

Unknown = NULL in relational database

Quiz Question

	V_CODE
1	21344
2	20001
3	24288
4	20001
5	24288

What row will be retrieved when the WHERE clause predicate is written as

V_CODE = 21344 AND V_CODE = 24288 ?

a. 1,3,5
b. 1
c. 3,5
d. No row will be retrieved

Recall predicates are evaluated for each row and the row is returned if the predicates evaluate to TRUE.

Quiz Question

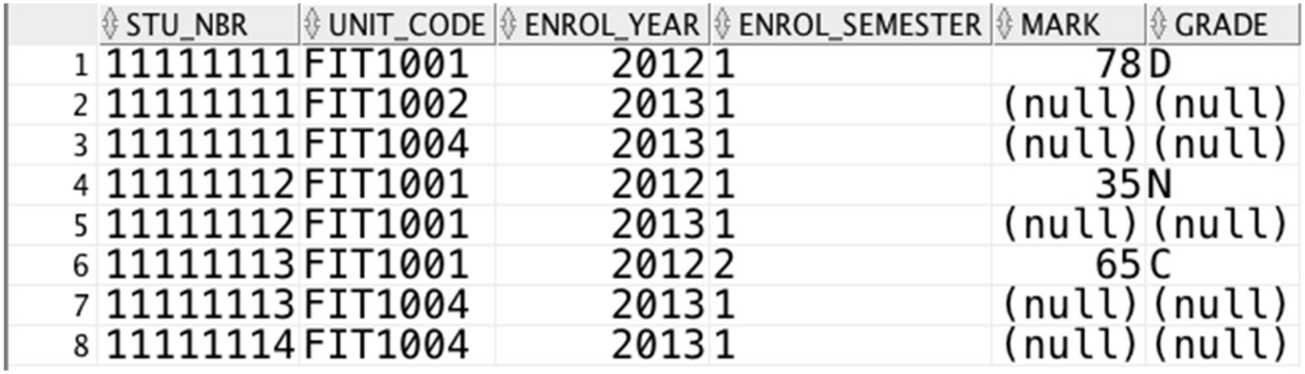
	V_CODE
1	21344
2	20001
3	24288
4	20001
5	24288

What rows will be retrieved when the WHERE clause predicate is written as

V_CODE <> 21344 OR V_CODE <> 24288 ?

a. 1,3,5
b. 2,4
c. 3,5
d. 1,2,3,4,5

Recall predicates are evaluated for each row and the row is returned if the predicates evaluate to TRUE.



	STU_NBR	UNIT_CODE	ENROL_YEAR	ENROL_SEMESTER	MARK	GRADE
1	11111111	FIT1001	2012	1	78	D
2	11111111	FIT1002	2013	1	(null)	(null)
3	11111111	FIT1004	2013	1	(null)	(null)
4	11111112	FIT1001	2012	1	35	N
5	11111112	FIT1001	2013	1	(null)	(null)
6	11111113	FIT1001	2012	2	65	C
7	11111113	FIT1004	2013	1	(null)	(null)
8	11111114	FIT1004	2013	1	(null)	(null)

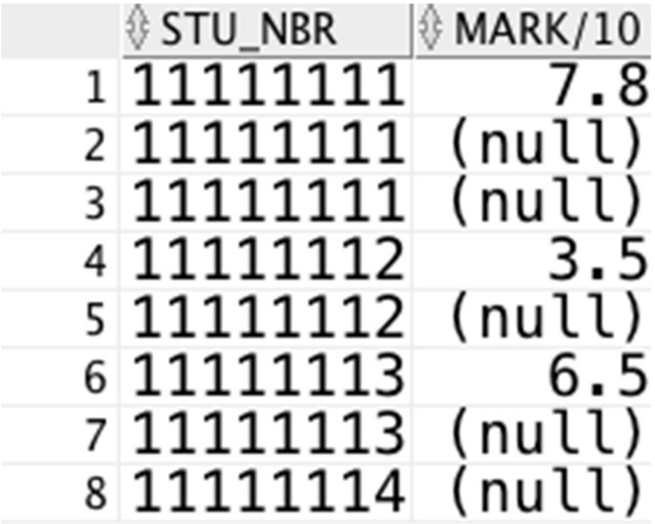
3. What is the correct SQL predicate to retrieve those students who have passed and also those students who have not been awarded any mark?

a. mark >= 50 AND mark IS NULL
b. mark >= 50 OR mark IS NULL
c. mark >= 50 AND mark IS NOT NULL
d. mark >= 50 OR mark IS NOT NULL
e. None of the above

Arithmetic Operations

- Can be performed in SQL.
- For example:

```
SELECT stu_nbr, mark/10
FROM enrolment;
```



	STU_NBR	MARK/10
1	11111111	7.8
2	11111111	(null)
3	11111111	(null)
4	11111112	3.5
5	11111112	(null)
6	11111113	6.5
7	11111113	(null)
8	11111114	(null)

Oracle NVL function

- It is used to replace a NULL with a value.

```
SELECT stu_nbr, NVL(mark,0), NVL(grade,’WH’)
          FROM enrolment;
```

Renaming Column

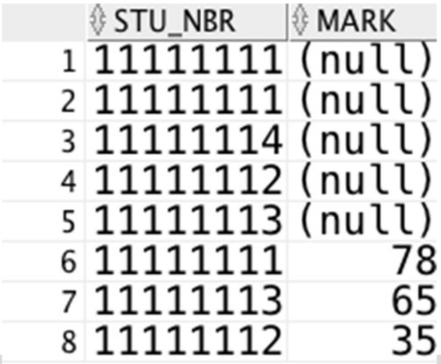
- Use the word “AS”
- Example

```
SELECT stu_nbr, mark/10 AS new_mark
FROM enrolment;
```

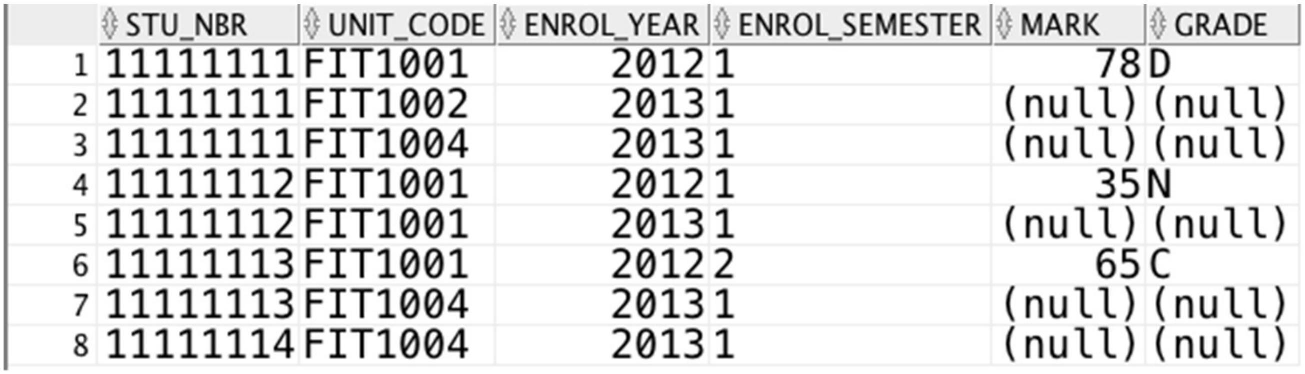
Sorting Query Result

- "ORDER BY" clause.
- Order can be ASCending or DESCending. The default is ASCending.
- NULL values can be placed first/last using "NULLS LAST" or "NULLS FIRST" command
- Sorting can be done for multiple columns.
  - order of the sorting is specified for each column.
- Example:

```
SELECT stu_nbr, mark
FROM enrolment
ORDER BY mark DESC
```



	STU_NBR	MARK
1	11111111	(null)
2	11111111	(null)
3	11111114	(null)
4	11111112	(null)
5	11111113	(null)
6	11111111	78
7	11111113	65
8	11111112	35



	STU_NBR	UNIT_CODE	ENROL_YEAR	ENROL_SEMESTER	MARK	GRADE
1	11111111	FIT1001	2012	1	78	D
2	11111111	FIT1002	2013	1	(null)	(null)
3	11111111	FIT1004	2013	1	(null)	(null)
4	11111112	FIT1001	2012	1	35	N
5	11111112	FIT1001	2013	1	(null)	(null)
6	11111113	FIT1001	2012	2	65	C
7	11111113	FIT1004	2013	1	(null)	(null)
8	11111114	FIT1004	2013	1	(null)	(null)

4. What will be the output of the following SQL statement?

```
SELECT stu_nbr
FROM enrolment
WHERE mark IS NULL;
```

a.

11111111
11111112
11111113
11111114

b.

11111111
11111111
11111112
11111113
11111114

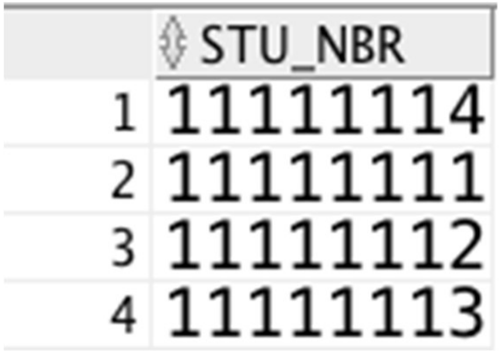
c.

11111111
11111112
11111113

Removing Duplicate Rows in the Query Result

- Use “DISTINCT” as part of SELECT clause.

```
SELECT DISTINCT stu_nbr
FROM enrolment
WHERE mark IS NULL;
```



	STU_NBR
1	11111114
2	11111111
3	11111112
4	11111113

SQL JOIN

STUDENT

sno	name
1	alex
2	maria
3	bob

```
SELECT *
FROM student JOIN qualification ON
student.sno = qualification.sno
```

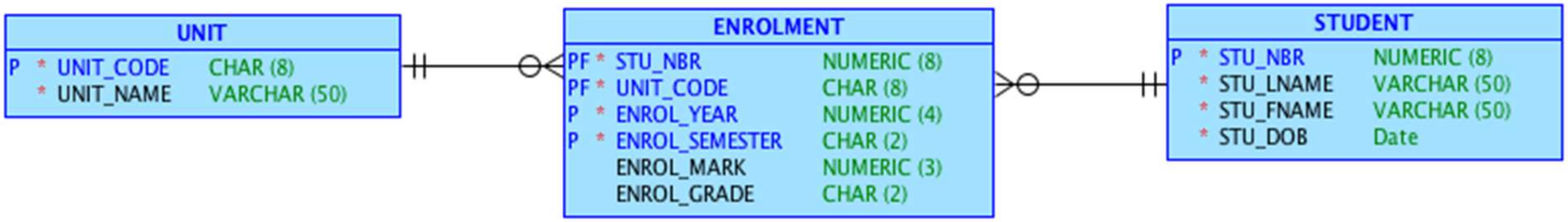
QUALIFICATION

sno	degree	year
1	bachelor	1990
1	master	2000
2	PhD	2001

sno	name	sno	degree	year
1	alex	1	bachelor	1990
1	alex	1	master	2000
2	maria	2	PhD	2001

JOIN-ing Multiple Tables

- Pair the PK and FK in the JOIN condition.



UNIT			
P	*	UNIT_CODE	CHAR (8)
	*	UNIT_NAME	VARCHAR (50)

ENROLMENT			
PF	*	STU_NBR	NUMERIC (8)
PF	*	UNIT_CODE	CHAR (8)
P	*	ENROL_YEAR	NUMERIC (4)
P	*	ENROL_SEMESTER	CHAR (2)
		ENROL_MARK	NUMERIC (3)
		ENROL_GRADE	CHAR (2)

STUDENT			
P	*	STU_NBR	NUMERIC (8)
	*	STU_LNAME	VARCHAR (50)
	*	STU_FNAME	VARCHAR (50)
	*	STU_DOB	Date

```
SELECT s.stu_nbr, s.stu_lname, u.unit_name
FROM unit u JOIN enrolment e ON u.unit_code=e.unit_code JOIN
student s ON e.stu_nbr=s.stu_nbr;
```

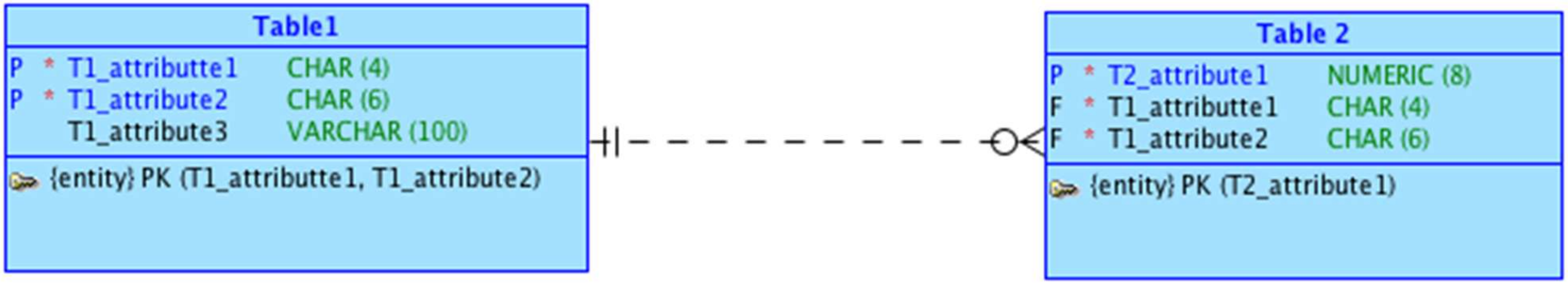


Table1

P	*	T1_attributte1	CHAR (4)
P	*	T1_attribute2	CHAR (6)
		T1_attribute3	VARCHAR (100)

{entity} PK (T1_attributte1, T1_attribute2)

Table 2

P	*	T2_attribute1	NUMERIC (8)
F	*	T1_attributte1	CHAR (4)
F	*	T1_attribute2	CHAR (6)

{entity} PK (T2_attribute1)

How many conditions will be used to join the two tables?

```
SELECT *
FROM table1 t1 JOIN table2 t2 ON
       (t1.T1_attribute1 = t2.T1_attribute1
         AND
         t1.T1_attribute2 = t2.T1_attribute2);
```

Summary

- SQL statement, clause, predicate.
- Writing SQL predicates.
  - Comparison, range, set membership, pattern matching, is NULL
  - Combining predicates using logic operators (AND, OR, NOT)
- Arithmetic operation.
  - NVL function
- Column alias.
- Sorting result.
- Removing duplicate rows.
- JOIN-ing tables

Oracle Date Data Type

- Dates are stored differently from the SQL standard
  - standard uses two different types: date and time
  - Oracle uses one type: DATE
    - Stored in internal format contains date and time
    - Output is controlled by formatting
      - select to_char(sysdate,'dd-Mon-yyyy') from dual;
        - » 01-Aug-2012
      - select to_char(sysdate,'dd-Mon-yyyy hh:mi:ss PM') from dual;
        - » 01-Aug-2012 10:56:50 AM

- DATE data type should be formatted with **TO_CHAR** when selecting for **display**.
- Text representing date must be formatted with **TO_DATE** when **comparing** or **inserting/updating.**
- Example:

```
select studid,
   studfname || ' ' || studlname as StudentName,
   to_char(studdob,'dd-Mon-yyyy') as StudentDOB
from uni.student
where studdob >  to_date('01-Apr-1991','dd-Mon-yyyy')
order by studdob;
```

Date Format

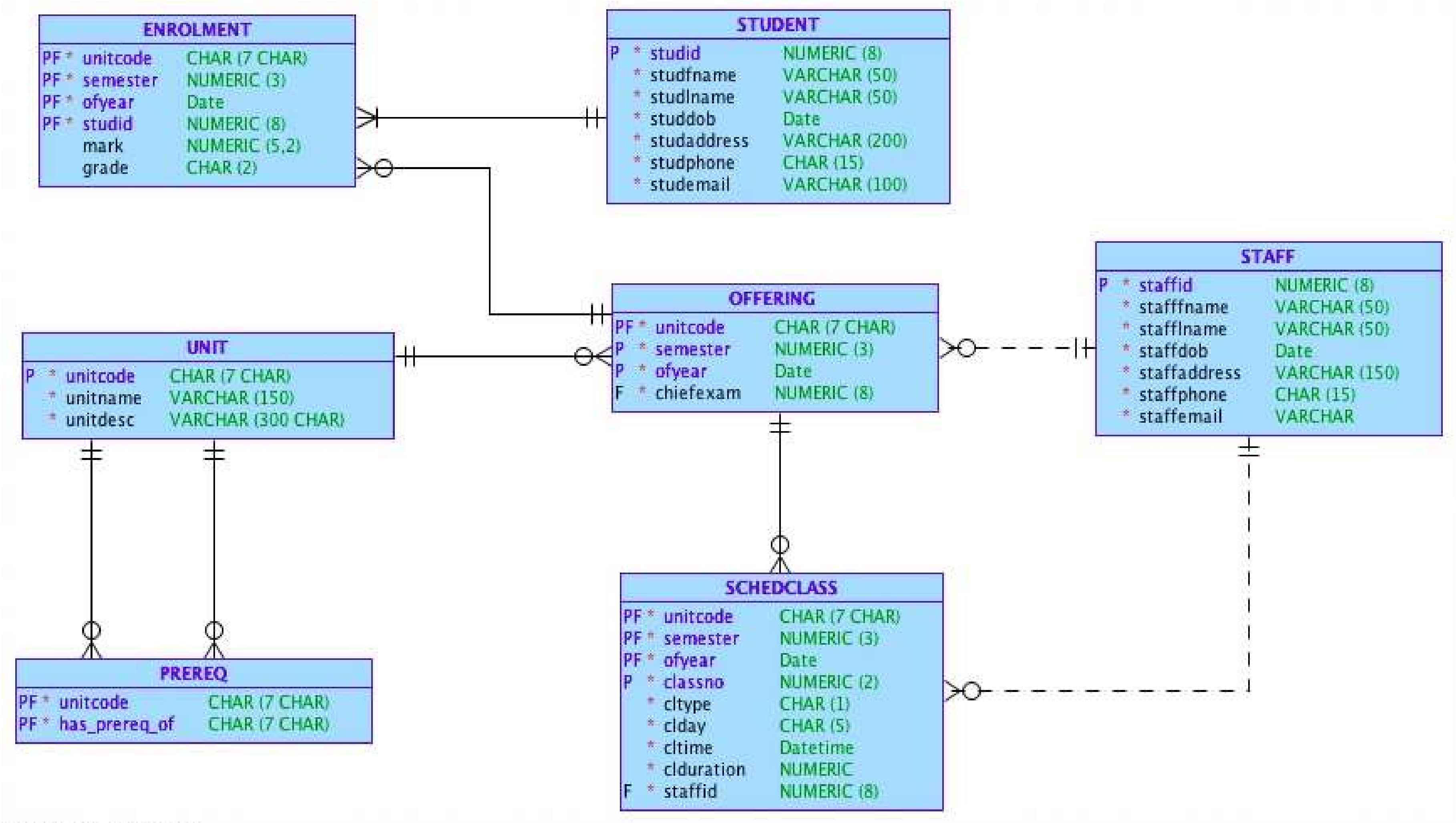
Oracle SQL Language Reference Manual
http://docs.oracle.com/cd/B19306_01/server.102/b14200/sql_elements004.htm
Example:

    ‘dd-mon-yyyy’ => ‘12-jan-2016’
     ‘dd-mm-yyyy’ => ‘12-01-2016’
    ‘dd/mm/yyyy’ => ‘12/1/2016’
    ‘dd-mon-yyyy hh24:mi’ => ‘12-jan-2016 14:15’

Current Date

- Current date can be queried from the DUAL table using the SYSDATE attribute.
  - SELECT sysdate FROM dual;
- Oracle internal attributes include:
  - sysdate: current date/time
  - systimestamp: current date/time as a timestamp
  - user: current logged in user

Practice



ENROLMENT

PF *	unitcode	CHAR (7 CHAR)
PF *	semester	NUMERIC (3)
PF *	ofyear	Date
PF *	studid	NUMERIC (8)
	mark	NUMERIC (5,2)
	grade	CHAR (2)

STUDENT

P *	studid	NUMERIC (8)
*	studfname	VARCHAR (50)
*	studlname	VARCHAR (50)
*	studdob	Date
*	studaddress	VARCHAR (200)
*	studphone	CHAR (15)
*	studemail	VARCHAR (100)

STAFF

P *	staffid	NUMERIC (8)
*	stafffname	VARCHAR (50)
*	stafflname	VARCHAR (50)
*	staffdob	Date
*	staffaddress	VARCHAR (150)
*	staffphone	CHAR (15)
*	staffemail	VARCHAR

OFFERING

PF *	unitcode	CHAR (7 CHAR)
P *	semester	NUMERIC (3)
P *	ofyear	Date
F *	chiefexam	NUMERIC (8)

UNIT

P *	unitcode	CHAR (7 CHAR)
*	unitname	VARCHAR (150)
*	unitdesc	VARCHAR (300 CHAR)

PREREQ

PF *	unitcode	CHAR (7 CHAR)
PF *	has_prereq_of	CHAR (7 CHAR)

SCHEDCLASS

PF *	unitcode	CHAR (7 CHAR)
PF *	semester	NUMERIC (3)
PF *	ofyear	Date
P *	classno	NUMERIC (2)
*	cltype	CHAR (1)
*	clday	CHAR (5)
*	cltime	Datetime
*	clduration	NUMERIC
F *	staffid	NUMERIC (8)

University data model