COMP 3311
Database Management Systems

Lab 3

Basic SQL Statements

Lab Topics

- The select-from-where SQL clauses.
- The order by clause.
- Simple join operations.

To see the result of the example queries in these lab notes, run them in SQL Developer against the database created by the Lab3DB.sql script file.

Lab 3 Database

Student table

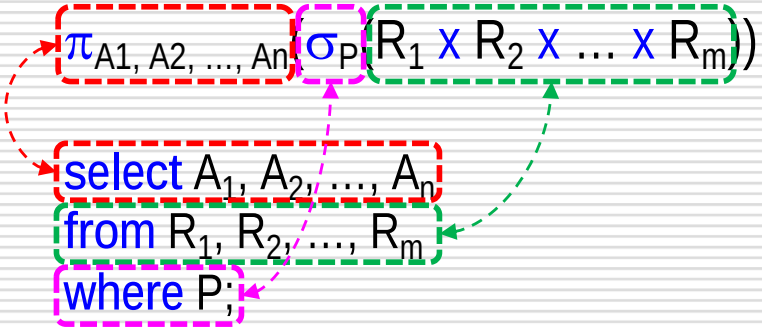
STUDENTID	FIRSTNAME	LASTNAME	EMAIL	PHONENO	CGA	DEPARTMENTID	ADMISSIONYEAR
13455789	Harry	Potter	cspotter	23581234	2.76	COMP	2017
15456789	Leonardo	Da Vinci	csdavinci	23585678	2.72	COMP	2017
13556789	Legolas	Greenleaf	magreenleaf	23582468	3.36	MATH	2018
13456789	Ariana	Grande	csgrande	23581234	2.82	COMP	2018
15678989	Maria	Callas	cscallas	23589876	2.73	COMP	2018
15678901	Albert	Einstein	cseinstein	23585678	2.56	COMP	2017
16789012	Robert	Redford	maredford	23582468	2.57	MATH	2018
14567890	Julius	Caesar	eecaesar	23589876	1.9	ELEC	2018
99987654	Lazzy	Lazy	cslazy	23581357		COMP	2018
26184624	Bruce	Wayne	eewayne	28261057	2.47	ELEC	2017
26184444	Donald	Trump	bstrump	28255057	1.49	BUS	2018
26186666	Warren	Buffet	bsbuffet	28266027	3.42	BUS	2017
66666666	Ferris	Bueller	bsbueller	28282727	1.64	BUS	2017
15000655	Steve	Jobs	csjobs	26232244	3.45	COMP	2017
15085942	Bill	Gates	csgates	25678679	3.4	COMP	2018
28834512	Issac	Newton	manewton	22861987	2.98	MATH	2017
28918856	Alan	Turing	maturing	26679834	3.56	MATH	2017
29873381	Nikola	Tesla	eetesla	25671983	3.37	ELEC	2017
13782973	Edith	Clarke	eeclarke	28340180	3.15	ELEC	2017
18792018	Elon	Musk	bsmusk	28659910	3.25	BUS	2018

Department table

DEPARTMENTID	DEPARTMENTNAME	ROOMNO
COMP	Computer Science	3528
MATH	Mathematics	3461
ELEC	Electronic Engineering	2528
BUS	Business	4528
HUMA	Humanities	1200

SELECT Statement

- Recall the correspondence between relational algebra and the select statement.



- Basic select statement syntax

```
select * { [distinct] column | expression [alias], ...}
from table, ...
where condition
order by column [asc | desc], ...;
```

SELECT Statement — Examples

- Retrieve all table columns.

```
select *
from Department;
```

DEPARTMENTID	DEPARTMENTNAME	ROOMNO
COMP	Computer Science	3528
MATH	Mathematics	3461
ELEC	Electronic Engineering	2528
BUS	Business	4528
HUMA	Humanities	1200

- Retrieve specified table columns.

```
select departmentId, departmentName
from Department;
```

DEPARTMENTID	DEPARTMENTNAME
COMP	Computer Science
MATH	Mathematics
ELEC	Electronic Engineering
BUS	Business
HUMA	Humanities

SELECT Statement — Removing Duplicate Results

- The default setting for the select statement is to return all the qualifying records – including duplicate ones.

- The statement on the right will return all the department ids from the Student table (20 values, one for each student).

```
select departmentId
from Student;
```

DEPARTMENTID
COMP
COMP
MATH
COMP
COMP
COMP
MATH
ELEC
COMP
ELEC
BUS
BUS
BUS
COMP
COMP
MATH
MATH
ELEC
ELEC
BUS

- To remove duplicates, the distinct keyword can be added to the select statement:

```
select distinct departmentId
from Student;
```

DEPARTMENTID
BUS
COMP
ELEC
MATH

Student(studentId, firstName, lastName, email, phoneNo, cga, departmentId, admissionYear)
Department(departmentId, departmentName, roomNo)

SELECT Statement — Incorporating Arithmetic Operations

- Arithmetic operations like * , / , + , - can be included in a select statement.

```
select lastName, cga, cga+2.0
from Student;
```

LASTNAME	CGA	CGA+2
Potter	2.76	4.76
Da Vinci	2.72	4.72
Greenleaf	3.36	5.36
Grande	2.82	4.82
Callas	2.73	4.73
Einstein	2.56	4.56
Redford	2.57	4.57
Caesar	1.9	3.9
Lazy	(null)	(null)
Wayne	2.47	4.47
Trump	1.49	3.49
Buffet	3.42	5.42
Bueller	1.64	3.64
Jobs	3.45	5.45
Gates	3.4	5.4
Newton	2.98	4.98
Turing	3.56	5.56
Tesla	3.37	5.37
Clarke	3.15	5.15
Musk	3.25	5.25

```
select lastName, cga, cga/2.0
from Student;
```

LASTNAME	CGA	CGA/2
Potter	2.76	1.38
Da Vinci	2.72	1.36
Greenleaf	3.36	1.68
Grande	2.82	1.41
Callas	2.73	1.365
Einstein	2.56	1.28
Redford	2.57	1.285
Caesar	1.9	0.95
Lazy	(null)	(null)
Wayne	2.47	1.235
Trump	1.49	0.745
Buffet	3.42	1.71
Bueller	1.64	0.82
Jobs	3.45	1.725
Gates	3.4	1.7
Newton	2.98	1.49
Turing	3.56	1.78
Tesla	3.37	1.685
Clarke	3.15	1.575
Musk	3.25	1.625

Note: cga/2.0 will return the same result as cga/2 in SQL, this is different from some higher-level languages like C++.

Student(studentId, firstName, lastName, email, phoneNo, cga, departmentId, admissionYear)
Department(departmentId, departmentName, roomNo)

SELECT Statement — Renaming A Query Result Column

- A column name in the query result can be renamed by using the as keyword.

```
select lastName as Familyname
from Student;
```

FAMIYNAME
Potter
Da Vinci
Greenleaf
⋮

- The select statement can be used to output a column named Quarter CGA which displays the result cga/4.

```
select lastName, cga/4 as "Quarter CGA"
from Student;
```

LASTNAME	Quarter CGA
Potter	0.69
Da Vinci	0.68
Greenleaf	0.84
⋮	

Note: Double quotes are required around an alias only if it has an embedded space as in the example above.

Student(studentId, firstName, lastName, email, phoneNo, cga, departmentId, admissionYear)
Department(departmentId, departmentName, roomNo)

SELECT Statement — Concatenating Query Results

- The || operator can be used to concatenate two columns in a select statement.

```
select firstName || ' ' || lastName as "Full Name"
from Student;
```

Full Name
Harry Potter
Leonardo Da Vinci
Legolas Greenleaf
⋮

- The || operator can be used to add a string to the result.

```
select firstName || ' ' || lastName || ' studies in ' || departmentId as "Description"
from Student;
```

Description
Harry Potter studies in COMP
Leonardo Da Vinci studies in COMP
Legolas Greenleaf studies in MATH
⋮

Note: If double quotes are placed around a single word alias such as Description, then it is displayed as typed; otherwise the alias name will be displayed in all capital letters.

SELECT Statement — Example Concatenation

- Concatenation can make a query result more comprehensible.

```
select firstName ||' '|| lastName || '(' || studentId || ') ' || 'from the ' || departmentId ||
  ' department has CGA ' || CGA ||'.' ||' His/Her email is ' || email || '@connect.ust.hk.'
  as lab3
from Student;
```

LAB3
Harry Potter(13455789) from the COMP Department has CGA 2.76. His/Her email is cspotter@connect.ust.hk.
Leonardo Da Vinci(15456789) from the COMP Department has CGA 2.72. His/Her email is csdavinci@connect.ust.hk.
Legolas Greenleaf(13556789) from the MATH Department has CGA 3.36. His/Her email is magreenleaf@connect.ust.hk.
Ariana Grande(13456789) from the COMP Department has CGA 2.82. His/Her email is csgrande@connect.ust.hk.
Maria Callas(15678989) from the COMP Department has CGA 2.73. His/Her email is cscallas@connect.ust.hk.
Albert Einstein(15678901) from the COMP Department has CGA 2.56. His/Her email is cseinstein@connect.ust.hk.
Robert Redford(16789012) from the MATH Department has CGA 2.57. His/Her email is maredford@connect.ust.hk.
Julius Caesar(14567890) from the ELEC Department has CGA 1.9. His/Her email is eecaesar@connect.ust.hk.
Lazzy Lazy(99987654) from the COMP Department has CGA . His/Her email is cslazy@connect.ust.hk.
Bruce Wayne(26184624) from the ELEC Department has CGA 2.47. His/Her email is eewayne@connect.ust.hk.
Donald Trump(26184444) from the BUS Department has CGA 1.49. His/Her email is bstrump@connect.ust.hk.
Warren Buffet(26186666) from the BUS Department has CGA 3.42. His/Her email is bsbuffet@connect.ust.hk.
Ferris Bueller(66666666) from the BUS Department has CGA 1.64. His/Her email is bsbueller@connect.ust.hk.
Steve Jobs(15000655) from the COMP Department has CGA 3.45. His/Her email is csjobs@connect.ust.hk.
Bill Gates(15085942) from the COMP Department has CGA 3.4. His/Her email is csgates@connect.ust.hk.
Issac Newton(28834512) from the MATH Department has CGA 2.98. His/Her email is manewton@connect.ust.hk.
Alan Turing(28918856) from the MATH Department has CGA 3.56. His/Her email is maturing@connect.ust.hk.
Nikola Tesla(29873381) from the ELEC Department has CGA 3.37. His/Her email is eetesla@connect.ust.hk.
Edith Clarke(13782973) from the ELEC Department has CGA 3.15. His/Her email is eeclarke@connect.ust.hk.
Elon Musk(18792018) from the BUS Department has CGA 3.25. His/Her email is bsmusk@connect.ust.hk.

Student(studentId, firstName, lastName, email, phoneNo, cga, departmentId, admissionYear)
Department(departmentId, departmentName, roomNo)

SELECT Statement — WHERE Clause

- The where clause restricts the select statement so that only specified rows from a table are retrieved.

```
select *
from Department
where departmentId='COMP';
```

DEPARTMENTID	DEPARTMENTNAME	ROOMNO
COMP	Computer Science	3528

The string 'COMP' in the condition clause is case sensitive.

- WHERE clause comparison operators:

=	equal	>	greater than	<	less than
>=	greater or equal	<=	less or equal	<>	not equal

Examples:

```
select *
from Student
where cga<>2.5;
```

```
select *
from Student
where cga<=1.9;
```

Student(studentId, firstName, lastName, email, phoneNo, cga, departmentId, admissionYear)
Department(departmentId, departmentName, roomNo)

WHERE Clause — Condition Operators (1)

- Range of values: **between** / **not between**

```
select *
from Student
where cga between 2.8 and 3;
```

STUDENTID	FIRSTNAME	LASTNAME	EMAIL	PHONENO	CGA	DEPARTMENTID	ADMISSIONYEAR
13456789	Ariana	Grande	csgrande	23581234	2.82	COMP	2018
28834512	Issac	Newton	manewton	22861987	2.98	MATH	2017

- Set membership: **in** / **not in**

```
select *
from Student
where departmentId in ('ELEC', 'MATH');
```

STUDENTID	FIRSTNAME	LASTNAME	EMAIL	PHONENO	CGA	DEPARTMENTID	ADMISSIONYEAR
13556789	Legolas	Greenleaf	magreenleaf	23582468	3.36	MATH	2018
16789012	Robert	Redford	maredford	23582468	2.57	MATH	2018
14567890	Julius	Caesar	eecaesar	23589876	1.9	ELEC	2018
26184624	Bruce	Wayne	eewayne	28261057	2.47	ELEC	2017
28834512	Issac	Newton	manewton	22861987	2.98	MATH	2017
28918856	Alan	Turing	maturing	26679834	3.56	MATH	2017
29873381	Nikola	Tesla	eetesla	25671983	3.37	ELEC	2017
13782973	Edith	Clarke	eeclarke	28340180	3.15	ELEC	2017

Student(studentId, firstName, lastName, email, phoneNo, cga, departmentId, admissionYear)
Department(departmentId, departmentName, roomNo)

WHERE Clause — Condition Operators (2)

- Null value: is null

```
select *
from Student
where cga is null;
```

STUDENTID	FIRSTNAME	LASTNAME	EMAIL	PHONENO	CGA	DEPARTMENTID	ADMISSIONYEAR
99987654	Lazzy	Lazy	cslazy	23581357	(null)	COMP	2018

NOTE: Cannot use where cga=null. This is illegal in SQL as null values are treated in a special way.

WHERE Clause — Boolean Operators (1)

- **and**

```
select *
from Student
where cga>=2 and departmentId='MATH';
```

STUDENTID	FIRSTNAME	LASTNAME	EMAIL	PHONENO	CGA	DEPARTMENTID	ADMISSIONYEAR
13556789	Legolas	Greenleaf	magreenleaf	23582468	3.36	MATH	2018
16789012	Robert	Redford	maredford	23582468	2.57	MATH	2018
28834512	Issac	Newton	manewton	22861987	2.98	MATH	2017
28918856	Alan	Turing	maturing	26679834	3.56	MATH	2017

- **not**

```
select *
from Student
where not departmentId='COMP';
```

STUDENTID	FIRSTNAME	LASTNAME	EMAIL	PHONENO	CGA	DEPARTMENTID	ADMISSIONYEAR
13556789	Legolas	Greenleaf	magreenleaf	23582468	3.36	MATH	2018
16789012	Robert	Redford	maredford	23582468	2.57	MATH	2018
14567890	Julius	Caesar	eecaesar	23589876	1.9	ELEC	2018
26184624	Bruce	Wayne	eewayne	28261057	2.47	ELEC	2017
26184444	Donald	Trump	bstrump	28255057	1.49	BUS	2018
26186666	Warren	Buffet	bsbuffet	28266027	3.42	BUS	2017
66666666	Ferris	Bueller	bsbueller	28282727	1.64	BUS	2017
28834512	Issac	Newton	manewton	22861987	2.98	MATH	2017
28918856	Alan	Turing	maturing	26679834	3.56	MATH	2017
29873381	Nikola	Tesla	eetesla	25671983	3.37	ELEC	2017
13782973	Edith	Clarke	eeclarke	28340180	3.15	ELEC	2017
18792018	Elon	Musk	bsmusk	28659910	3.25	BUS	2018

Student(studentId, firstName, lastName, email, phoneNo, cga, departmentId, admissionYear)
Department(departmentId, departmentName, roomNo)

WHERE Clause — Boolean Operators (2)

- **or**

```
select *
from Student
where cga>=2 or departmentId='MATH';
```

STUDENTID	FIRSTNAME	LASTNAME	EMAIL	PHONENO	CGA	DEPARTMENTID	ADMISSIONYEAR
13455789	Harry	Potter	cspotter	23581234	2.76	COMP	2017
15456789	Leonardo	Da Vinci	csdavinci	23585678	2.72	COMP	2017
13556789	Legolas	Greenleaf	magreenleaf	23582468	3.36	MATH	2018
13456789	Ariana	Grande	csgrande	23581234	2.82	COMP	2018
15678989	Maria	Callas	cscallas	23589876	2.73	COMP	2018
15678901	Albert	Einstein	cseinstein	23585678	2.56	COMP	2017
16789012	Robert	Redford	maredford	23582468	2.57	MATH	2018
26184624	Bruce	Wayne	eewayne	28261057	2.47	ELEC	2017
26186666	Warren	Buffet	bsbuffet	28266027	3.42	BUS	2017
15000655	Steve	Jobs	csjobs	26232244	3.45	COMP	2017
15085942	Bill	Gates	csgates	25678679	3.4	COMP	2018
28834512	Issac	Newton	manewton	22861987	2.98	MATH	2017
28918856	Alan	Turing	maturing	26679834	3.56	MATH	2017
29873381	Nikola	Tesla	eetesla	25671983	3.37	ELEC	2017
13782973	Edith	Clarke	eeclarke	28340180	3.15	ELEC	2017
18792018	Elon	Musk	bsmusk	28659910	3.25	BUS	2018

WHERE Clause — Boolean Operator Precedence (1)

- The **and** operator has higher precedence than the **or** operator.
  - Select students from the COMP department *plus* also students from the MATH department with cga>=3:

```
select *
from Student
where departmentId='COMP' or departmentId='MATH' and cga>=3;
```

STUDENTID	FIRSTNAME	LASTNAME	EMAIL	PHONENO	CGA	DEPARTMENTID	ADMISSIONYEAR
13455789	Harry	Potter	cspotter	23581234	2.76	COMP	2017
15456789	Leonardo	Da Vinci	csdavinci	23585678	2.72	COMP	2017
13556789	Legolas	Greenleaf	magreenleaf	23582468	3.36	MATH	2018
13456789	Ariana	Grande	csgrande	23581234	2.82	COMP	2018
15678989	Maria	Callas	cscallas	23589876	2.73	COMP	2018
15678901	Albert	Einstein	cseinstein	23585678	2.56	COMP	2017
99987654	Lazzy	Lazy	cslazy	23581357		COMP	2018
15000655	Steve	Jobs	csjobs	26232244	3.45	COMP	2017
15085942	Bill	Gates	csgates	25678679	3.4	COMP	2018
28918856	Alan	Turing	maturing	26679834	3.56	MATH	2017

Student(studentId, firstName, lastName, email, phoneNo, cga, departmentId, admissionYear)
Department(departmentId, departmentName, roomNo)

WHERE Clause — Boolean Operator Precedence (2)

- Use parenthesis to change the precedence order.
  - To select students with cga>=3, from either the 'COMP' or the 'MATH' departments, add parentheses.

```
select *
from Student
where (departmentId='COMP' or departmentId='MATH') and cga>=3;
```

STUDENTID	FIRSTNAME	LASTNAME	EMAIL	PHONENO	CGA	DEPARTMENTID	ADMISSIONYEAR
13556789	Legolas	Greenleaf	magreenleaf	23582468	3.36	MATH	2018
15000655	Steve	Jobs	csjobs	26232244	3.45	COMP	2017
15085942	Bill	Gates	csgates	25678679	3.4	COMP	2018
28918856	Alan	Turing	maturing	26679834	3.56	MATH	2017

WHERE Clause — String Matching (1)

- **like** (for matching patterns)

  % can match zero or more characters.

  Find students whose first name contains a "a" anywhere in the name.

What is the query result?

```
select *
from Student
where firstName like '%a%';
```

```
select *
from Student
where firstName like 'A%';
```

STUDENTID	FIRSTNAME	LASTNAME	EMAIL	PHONENO	CGA	DEPARTMENTID	ADMISSIONYEAR
13455789	Harry	Potter	cspotter	23581234	2.76	COMP	2017
15456789	Leonardo	Da Vinci	csdavinci	23585678	2.72	COMP	2017
13556789	Legolas	Greenleaf	magreenleaf	23582468	3.36	MATH	2018
13456789	Ariana	Grande	csgrande	23581234	2.82	COMP	2018
15678989	Maria	Callas	cscallas	23589876	2.73	COMP	2018
99987654	Lazzy	Lazy	cslazy	23581357		COMP	2018
26184444	Donald	Trump	bstrump	28255057	1.49	BUS	2018
26186666	Warren	Buffet	bsbuffet	28266027	3.42	BUS	2017
28834512	Issac	Newton	manewton	22861987	2.98	MATH	2017
28918856	Alan	Turing	maturing	26679834	3.56	MATH	2017
29873381	Nikola	Tesla	eetesla	25671983	3.37	ELEC	2017

Student(studentId, firstName, lastName, email, phoneNo, cga, departmentId, admissionYear)
Department(departmentId, departmentName, roomNo)

WHERE Clause — String Matching (2)

- **like** (for matching patterns)

  _ (underscore) matches exactly one character.

  Find students whose first name contains a "u" as the second character.

What is the query result?

```
select *
from Student
where firstName like '_u%';
```

```
select *
from Student
where firstName like '%_u%';
```

STUDENTID	FIRSTNAME	LASTNAME	EMAIL	PHONENO	CGA	DEPARTMENTID	ADMISSIONYEAR
14567890	Julius	Caesar	eecaesar	23589876	1.9	ELEC	2018

WHERE Clause — String Matching (3)

- regexp_like function (for matching regular expressions)

  Syntax: regexp_like(*attribute-name*, *regular-expression*, *match-parameter*)

  match-parameter: i → case insensitive; c → case sensitive.

  Find students with a double vowel in their last name.

```
select *
from Student
where regexp_like(lastName, '([aeiou])\1', 'i');
```

STUDENTID	FIRSTNAME	LASTNAME	EMAIL	PHONENO	CGA	DEPARTMENTID	ADMISSIONYEAR
13556789	Legolas	Greenleaf	magreenleaf	23582468	3.36	MATH	2018

For information on the regular expressions supported by Oracle see https://docs.oracle.com/cd/B12037_01/server.101/b10759/ap_posix001.htm#i690819.

For examples of regular expression use in Oracle see https://www.salvis.com/blog/2018/09/28/regular-expressions-sql-examples/.

For testing your regular expressions see https://www.regextester.com/ (use the PCRE option).

ORDER BY Clause (1)

- **asc** ascending order (default)

```
select studentId, firstName, LastName, cga
from Student
where departmentId='COMP'
order by cga;
```

STUDENTID	FIRSTNAME	LASTNAME	CGA
15678901	Albert	Einstein	2.56
15456789	Leonardo	Da Vinci	2.72
15678989	Maria	Callas	2.73
13455789	Harry	Potter	2.76
13456789	Ariana	Grande	2.82
15085942	Bill	Gates	3.4
15000655	Steve	Jobs	3.45
99987654	Lazzy	Lazy	(null)

- **desc** descending order

```
select studentId, firstName, LastName, cga
from Student
where departmentId='COMP'
order by cga desc;
```

STUDENTID	FIRSTNAME	LASTNAME	CGA
99987654	Lazzy	Lazy	(null)
15000655	Steve	Jobs	3.45
15085942	Bill	Gates	3.4
13456789	Ariana	Grande	2.82
13455789	Harry	Potter	2.76
15678989	Maria	Callas	2.73
15456789	Leonardo	Da Vinci	2.72
15678901	Albert	Einstein	2.56

Student(studentId, firstName, lastName, email, phoneNo, cga, departmentId, admissionYear)
Department(departmentId, departmentName, roomNo)

ORDER BY Clause (2)

- Sort on multiple columns

```
select *
from Student
order by departmentId asc, lastName desc;
```

STUDENTID	FIRSTNAME	LASTNAME	EMAIL	PHONENO	CGA	DEPARTMENTID	ADMISSIONYEAR
26184444	Donald	Trump	bstrump	28255057	1.49	BUS	2018
18792018	Elon	Musk	bsmusk	28659910	3.25	BUS	2018
26186666	Warren	Buffet	bsbuffet	28266027	3.42	BUS	2017
66666666	Ferris	Bueller	bsbueller	28282727	1.64	BUS	2017
13455789	Harry	Potter	cspotter	23581234	2.76	COMP	2017
99987654	Lazzy	Lazy	cslazy	23581357		COMP	2018
15000655	Steve	Jobs	csjobs	26232244	3.45	COMP	2017
13456789	Ariana	Grande	csgrande	23581234	2.82	COMP	2018
15085942	Bill	Gates	csgates	25678679	3.4	COMP	2018
15678901	Albert	Einstein	cseinstein	23585678	2.56	COMP	2017
15456789	Leonardo	Da Vinci	csdavinci	23585678	2.72	COMP	2017
15678989	Maria	Callas	cscallas	23589876	2.73	COMP	2018
26184624	Bruce	Wayne	eewayne	28261057	2.47	ELEC	2017
29873381	Nikola	Tesla	eetesla	25671983	3.37	ELEC	2017
13782973	Edith	Clarke	eeclarke	28340180	3.15	ELEC	2017
14567890	Julius	Caesar	eecaesar	23589876	1.9	ELEC	2018
28918856	Alan	Turing	maturing	26679834	3.56	MATH	2017
16789012	Robert	Redford	maredford	23582468	2.57	MATH	2018
28834512	Issac	Newton	manewton	22861987	2.98	MATH	2017
13556789	Legolas	Greenleaf	magreenleaf	23582468	3.36	MATH	2018

Student(studentId, firstName, lastName, email, phoneNo, cga, departmentId, admissionYear)
Department(departmentId, departmentName, roomNo)

Cartesian Product

- Cartesian product combines each row of one table with every row of another table.

```
select firstName, lastName, departmentName
from Student, Department;
```

Note: If the Student table has 20 entries and the Department table has 5 entries, then the query result has (20x5) 100 entries.

FIRSTNAME	LASTNAME	DEPARTMENTNAME
Harry	Potter	Computer Science
Leonardo	Da Vinci	Computer Science
Legolas	Greenleaf	Computer Science
Ariana	Grande	Computer Science
Maria	Callas	Computer Science
Albert	Einstein	Computer Science
Robert	Redford	Computer Science
Julius	Caesar	Computer Science
Lazzy	Lazy	Computer Science
Bruce	Wayne	Computer Science
Donald	Trump	Computer Science
Warren	Buffet	Computer Science
Ferris	Bueller	Computer Science
Steve	Jobs	Computer Science
Bill	Gates	Computer Science
Issac	Newton	Computer Science
Alan	Turing	Computer Science
Nikola	Tesla	Computer Science
Edith	Clarke	Computer Science
Elon	Musk	Computer Science
Harry	Potter	Mathematics
Leonardo	Da Vinci	Mathematics
Legolas	Greenleaf	Mathematics
Ariana	Grande	Mathematics
Maria	Callas	Mathematics
Albert	Einstein	Mathematics
Robert	Redford	Mathematics
Julius	Caesar	Mathematics
Lazzy	Lazy	Mathematics
Bruce	Wayne	Mathematics
Donald	Trump	Mathematics
Warren	Buffet	Mathematics
Ferris	Bueller	Mathematics
Steve	Jobs	Mathematics
Bill	Gates	Mathematics
Issac	Newton	Mathematics
Alan	Turing	Mathematics
Nikola	Tesla	Mathematics
Edith	Clarke	Mathematics
Elon	Musk	Mathematics
Harry	Potter	Electronic Engineering
Leonardo	Da Vinci	Electronic Engineering
Legolas	Greenleaf	Electronic Engineering
⋮		
Elon	Musk	Business
Harry	Potter	Humanities
Leonardo	Da Vinci	Humanities
Legolas	Greenleaf	Humanities
Ariana	Grande	Humanities
Maria	Callas	Humanities
Albert	Einstein	Humanities
Robert	Redford	Humanities
Julius	Caesar	Humanities
Lazzy	Lazy	Humanities
Bruce	Wayne	Humanities
Donald	Trump	Humanities
Warren	Buffet	Humanities
Ferris	Bueller	Humanities
Steve	Jobs	Humanities
Bill	Gates	Humanities
Issac	Newton	Humanities
Alan	Turing	Humanities
Nikola	Tesla	Humanities
Edith	Clarke	Humanities
Elon	Musk	Humanities

Student(studentId, firstName, lastName, email, phoneNo, cga, departmentId, admissionYear)
Department(departmentId, departmentName, roomNo)

Join (1)

- Join is a Cartesian product followed by a selection.

```
select firstName, lastName, departmentName
from Student, Department
where Student.departmentId=Department.departmentId;
```

FIRSTNAME	LASTNAME	DEPARTMENTNAME
Harry	Potter	Computer Science
Leonardo	Da Vinci	Computer Science
Legolas	Greenleaf	Mathematics
Ariana	Grande	Computer Science
Maria	Callas	Computer Science
Albert	Einstein	Computer Science
Robert	Redford	Mathematics
Julius	Caesar	Electronic Engineering
Lazzy	Lazy	Computer Science
Bruce	Wayne	Electronic Engineering
Donald	Trump	Business
Warren	Buffet	Business
Ferris	Bueller	Business
Steve	Jobs	Computer Science
Bill	Gates	Computer Science
Issac	Newton	Mathematics
Alan	Turing	Mathematics
Nikola	Tesla	Electronic Engineering
Edith	Clarke	Electronic Engineering
Elon	Musk	Business

Note: Attributes names need to be qualified with table names if they are ambiguous. For example, departmentId is an attribute of both the Student and Department tables in the above example.

If the Student table has 20 entries and the Department table has 5 entries, how many entries are there in the query result?

Student(studentId, firstName, lastName, email, phoneNo, cga, departmentId, admissionYear)
Department(departmentId, departmentName, roomNo)

Join (2)

- A join can also be specified as follows.

```
select firstName, lastName, departmentName
from Student join Department on Student.departmentId=Department.departmentId;
```

FIRSTNAME	LASTNAME	DEPARTMENTNAME
Harry	Potter	Computer Science
Leonardo	Da Vinci	Computer Science
Legolas	Greenleaf	Mathematics
Ariana	Grande	Computer Science
Maria	Callas	Computer Science
Albert	Einstein	Computer Science
Robert	Redford	Mathematics
Julius	Caesar	Electronic Engineering
Lazzy	Lazy	Computer Science
Bruce	Wayne	Electronic Engineering
Donald	Trump	Business
Warren	Buffet	Business
Ferris	Bueller	Business
Steve	Jobs	Computer Science
Bill	Gates	Computer Science
Issac	Newton	Mathematics
Alan	Turing	Mathematics
Nikola	Tesla	Electronic Engineering
Edith	Clarke	Electronic Engineering
Elon	Musk	Business

Join With Conditions

- Additional conditions in the **where** clause along with a join condition further restricts the tuples selected.

```
select *
from Student, Department
where Student.departmentId=Department.departmentId
        and Student.departmentId='COMP'
        and cga>2.5;
```

Note: The join attribute, departmentId, repeats in the query result.

STUDENTID	FIRSTNAME	LASTNAME	EMAIL	PHONENO	CGA	DEPARTMENTID	ADMISSIONYEAR	DEPARTMENTID_1	DEPARTMENTNAME	ROOMNO
13455789	Harry	Potter	cspotter	23581234	2.76	COMP	2017	COMP	Computer Science	3528
15456789	Leonardo	Da Vinci	csdavinci	23585678	2.72	COMP	2017	COMP	Computer Science	3528
13456789	Ariana	Grande	csgrande	23581234	2.82	COMP	2018	COMP	Computer Science	3528
15678989	Maria	Callas	cscallas	23589876	2.73	COMP	2018	COMP	Computer Science	3528
15678901	Albert	Einstein	cseinstein	23585678	2.56	COMP	2017	COMP	Computer Science	3528
15000655	Steve	Jobs	csjobs	26232244	3.45	COMP	2017	COMP	Computer Science	3528
15085942	Bill	Gates	csgates	25678679	3.4	COMP	2018	COMP	Computer Science	3528

Student(studentId, firstName, lastName, email, phoneNo, cga, departmentId, admissionYear)
Department(departmentId, departmentName, roomNo)

Natural Join

- A natural join merges the rows of the tables if columns with identical names match on their values and keeps only one of the join columns.
- For the tables Student and Department, only rows with identical values in the column departmentId will be merged, so students with departmentId ='COMP' will merge with the department with departmentId = 'COMP'.

```
select *
from Student natural join Department
where departmentId='COMP'
        and cga>2.5;
```

Note: The join (common) attribute, departmentId, does not repeat in the query result.

DEPARTMENTID	STUDENTID	FIRSTNAME	LASTNAME	EMAIL	PHONENO	CGA	ADMISSIONYEAR	DEPARTMENTNAME	ROOMNO
COMP	13455789	Harry	Potter	cspotter	23581234	2.76	2017	Computer Science	3528
COMP	15456789	Leonardo	Da Vinci	csdavinci	23585678	2.72	2017	Computer Science	3528
COMP	13456789	Ariana	Grande	csgrande	23581234	2.82	2018	Computer Science	3528
COMP	15678989	Maria	Callas	cscallas	23589876	2.73	2018	Computer Science	3528
COMP	15678901	Albert	Einstein	cseinstein	23585678	2.56	2017	Computer Science	3528
COMP	15000655	Steve	Jobs	csjobs	26232244	3.45	2017	Computer Science	3528
COMP	15085942	Bill	Gates	csgates	25678679	3.4	2018	Computer Science	3528

Student(studentId, firstName, lastName, email, phoneNo, cga, departmentId, admissionYear)
Department(departmentId, departmentName, roomNo)

The dual Table

- dual is an Oracle built-in table for SQL queries that do not logically have table names.

```
select 'The results of the queries are: '
from dual;
```

will output the string: The results of the queries are:

Note: To suppress the output of table column headers in the Script Output pane of SQL Developer, place the SQL*Plus command "set heading off" in a script file before the SQL statement(s) whose result column headers you want to suppress. Use the command "set heading on" to again show the column headers for the result of SQL statements.