



### DDL

CREATE TABLE tablename (column- name type column- restriction, columnname2 type columnrestriction, ...);	creates a table
CREATE INDEX name ON tablename- (column asc, column desc,...);	explicit creation of index (for efficiency for ex). Unique and primary keys will automatically create indexes!
DROP TABLE tablename	deletes tables if there are no references to thi table ou if these specify ON DELETE CASCADE. In this case, it deletes the table and all the reference lines on the other tables that refer to the deleted table
CREATE VIEW viewname (column- name type column- restriction, ...);	creates a view that can be used as a table

**Types:** varchar2(n) = string of n characters variable size 1<4000, char(n) = string of n characters fixed size 1<255, number(p,s), date, timestamp...

**Column restrictions:** none, primary key, not null, unique, references, check. **Table restrictions:** primary key(col, col...), foreign key(col, col...), check, references. All these depend on the db.

### DML

Insertion	INSERT INTO tablename VALUES- (val, val, val...)	adds a line with all the values in the specified order
	INSERT INTO tablen- am(col,col...) VALUES(val,val...)	adds a line only with the values for the specified columns
Modifi- cation	UPDATE tablename SET col1=expr1, col2=expr2 WHERE cond	all the lines that meet the cond have the col1 and col2 updated according to the expr1 and expr2

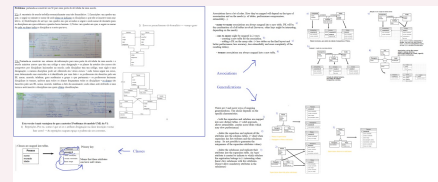
### DML (cont)

Deletion	DELETE FROM tablename WHERE cond	deletes all the line in the table that meet the cond
----------	--	--

The changes stay in a temporary state. To **commit** them permanently execut COMMIT. To **undo** the changes after the last commit do ROLLBACK.

It's possible to create **sequences** to automatically create values. Eg: create sequence num\_socio start with 1000 increment by 10; insert into sócios values( num\_socio.nextval, 'Quim' ); select num\_socio.currval from dual; --> Crie uma sequência para gerar automaticamente números de sócios

### UML to SQL



### Operators, Patterns & Symbols

+	plus
-	minus
*	times
/	divided
	concatenation
=	equal to
<>	different
!=	different
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to
[not] in	belongs [doesn't belong]
[not] between x and y	x <= value <= y [not]
x [not] like y	compares x to y
is [not] null	is[n't] null



### Operators, Patterns & Symbols (cont)

not	not
and	and
or	or
*	everything/all
_	any letter (only 1)
%	any sequence of characters
( )	fits queries inside other queries
distinct	eliminates duplicate rows

' ' - use for words

'M%' = Marina, M...

'M\_r%' = Mar, Mari, Moreira...

'a\_\_' = ant, add, alc...

On Microsoft Access use \* (instead of %) and ? (instead of \_)

#### Order of precedence:

1. Arithmetic operators (+ and - > \* and / > ||)
2. Comparasion operators
3. Logic operators (not > and > or)

( ) --> SELEC by, salario FROM orienta WHERE salario = (SELECT max(salario) FROM orienta);

### DQL Basics

SELECT rowname(s) FROM table name	displays all the info from the table on the row(s)
SELECT x FROM y WHERE anycriteria	displays all the x info, from table y, that meets the criteria
SELECT x FROM y WHERE criteria1 AND criteria2	diplays all the x info from table y, that meets the criteria 1 and 2
SELECT x, j FROM y ORDER BY j	displays all the x and j row's info, from y table, ordered by j
SELECT x, i FROM y GROUP BY x	displays the x and i info from table y, organized by x groups

### DQL Basics (cont)

SELECT x, i FROM y  
GROUP BY x HAVING  
criteria

displays the x and i info from table y that fits the criteria, organized by the x. criteria

ORDER BY applies to strings (alphabetically) and numbers (asc), and applies for more than one rows. Use **desc** to order backwards (SELECT x, j, i FROM y ORDER BY i, j desc).

GROUP BY organizes rows by a specific column.

**Example:** SELECT id, avg(classification) as grade FROM students GROUP BY id --> will calculate the average classification by id, taking that into account for the result on the grades column for ids that appear more than once.

### DQL Simple calculations

SELECT avg(column-name) as newcolumnname FROM tablename

displays the average result of the numbers in the column of the table chosen in a new columns called newcolumnname

SELECT count

displays the number of rows on columnname

SELECT sum

displays the addition of the numbers on the row

SELECT max

displays the higher number on the column

SELECT min

displays the smaller number

All of these can be used together (SELECT avg(x) as newname1, max(x) as newname2 FROM tablename);.

These are useful for as an example finding the average (avg) of a column, to count the total of rows of a column (count), the total of the values (sum) and the max and min numbers.



### DQL - others

rownum	n. of the row for the resulting table
rowid	internal address for the row/line on the db
case --- when --- else - -- end as	turns quantitative results into qualitative
nvl(valueX, valueY- fNule)	returns 'valueX' if it's not null and valueY- fNule if valueX is null

eg: SELECT rownum, rowid, column1, column3 FROM table; and "- SELECT columnname, column2name, CASE column3name WHEN n. THEN 'expression' WHEN othern. THEN 'otherexpression' ELSE 'anotherexpression' END AS newcolumnname FROM table; **Rownum** limits results to the first n lines for extensive outputs, while **rowid** allows quick access but is affected by import/export operations. **NVL** is also used as NVL(t, s, n), returning S if T is positive, otherwise N.

### DCL

GRANT privilegetype (col1, col2) ON tablename TO username WITH grantoption

Types of privilege: alter, delete, execute, index, insert, read, references, select, update, create session, alter session, drop any table.

Thre apply to tables, viws, sequences, functions, packages, system and/or users.

### Tehcnical support position

What type of queries are the most common on a technical support role? In this role, the most commonly used queries often involve retrieving and updating information related to users, tickets, issues, and system logs; data retrieval and correction; account management; configuration changes; audit trail analysis; performance tunign; report generation; data import/export issues. Egs:

1- Retrieve ticket information: SELECT \* FROM tickets WHERE ticket\_id = 'XYZ'

2- Updtate ticket status: UPDATE tickets SET status = 'closed' WHERE ticket\_id = 'XYZ'

### Tehcnical support position (cont)

- Review system logs to identify patterns or issues affecting multiple users. SELECT \* FROM system\_logs WHERE log\_type = 'Error' ORDER BY timestamp DESC LIMIT 10
- Track user activity and interactions withthe system for troubleshooting purposes. SELECT \* FROM user\_activity WHERE user\_id = 'ABC' ORDER BY timestamp DESC LIMIT 10
- Update user information. UPDATE users SET email = 'new\_email@example.com' WHERE user\_id = 'ABC'
- Check the status of a service. SELECT \* FROM service\_sttaus WHERE status = 'Down';
- Retrieve FAQ information from a knowledge base or faq database to provide quick solutions to common issues. SELECT \* FROM faq WHERE category ='Triubleshooting'.
- User authentication issues: check if user's credentials are valid. SELECT \* FROM users WHERE username ='user123' AND password= ' hashed\_password'
- Reset user passwords. UPDATE users SET password = 'new\_hshed\_password' WHERE username = 'user123'
- Check system resource usage: monitor resource usage to identify potencial performance issues. SELECT \* FROM system\_resources WHERE resource\_type = 'cpu' AND usage\_percentage > 90;
- Check recent system updates. SELECT \* FROM system\_updates ORDER BY update\_date DESC LIMIT 10



By **idd**  
[cheatography.com/idd/](https://cheatography.com/idd/)

Not published yet.  
Last updated 29th December, 2023.  
Page 4 of 4.

Sponsored by **ApolloPad.com**  
Everyone has a novel in them. Finish Yours!  
<https://apollopad.com>