

SQL Commands

COMMANDO	WERKING
SELECT <element> FROM <tabel>	Kolom uit tabel selecteren → SELECT zoekt de kolommen (gescheiden met komma) → FROM zoekt de tabellen die deze kolommen bevatten
SELECT * FROM <tabel>	Geeft de hele tabel weer
;	Aan einde van commando → uitvoeren
RUN	Laatste commando uitvoeren
R	
/	
SAVE <bestandsnaam>	Script opslaan (in bin)
START <bestandsnaam>	Script uitvoeren
EDIT	Open laatste script in kladblok (achteraf bestand>afsluiten) → Slaat het bestand in buffer op, moet nog permanent opslaan
SAVE <bestand> APPEND	Voegt script in buffer aan einde van bestand toe
DESC <naam tabel>	Geeft de datastructuur van een tabel weer (datatype, lengte,...)
SET LINESIZE <getal>	Bepaalt de hoeveelheid karakters die men op 1 lijn kan tonen
<tabel> + <getal>	Bv. salary + 300, geeft in tabel de resultaten van de som
-	Bv. salary - 300, geeft in tabel de resultaten van de aftrekking
*	Bv. salary * 3, geeft in tabel de resultaten van de vermenigvuldiging
/	Bv. salary / 3, geeft in tabel de resultaten van de deling
()	Beïnvloeden volgorde van bewerking (haakjes gaan eerst)
<naam kolom> "naam"	Geeft de kolom weer met als naam (vanboven) de gegeven naam → bv. bij bewerking
<kolom> <kolom> AS "naam"	Voegt inhoud van 2 kolommen samen onder gegeven naam → Output: kolom1_kolom2
<kolom> 'tekst' <kolom> AS "naam"	Voegt inhoud van 2 kolommen samen met daartussen 'tekst' en onder de hoofding "naam" → Output: kolom1 tekst kolom2
SELECT DISTINCT <element> FROM <tabel>	Geeft geen herhalingen van query weer → vb. SELECT DISTINCT department_id → geeft elke id maar 1 keer → DISTINCT moet bij <element> tussen haakjes wanneer er haakjes zijn
DESCRIBE <tabel>	Geeft opmaak van tabel weer (wat waar staat, grenzen van variabelen, nul of niet,...)
SELECT ... FROM ... WHERE <condition>	Geef een conditie mee om specifieker te zoeken → bv. WHERE last_name = 'Ceux'
SELECT SYSDATE FROM DUAL	From dual → toont resultaat van een bewerking (hier: sysdate) → Bestaat uit één cel
%	Vervangt 0 of meer karakters [Wildcard]
WHERE ... LIKE 'S%'	Waar de string met een S begint
_	Vervangt exact 1 karakter [Wildcard]
WHERE ... IS NULL	Waar de waarde van de variabele null is

WHERE ... AND ...	Geef meerdere condities mee
&<te vervangen variabele>	Prompt de gebruiker om een waarde in te geven
UPPER ('&string')	Zet de ingegeven string in hoofdletters
WHERE UPPER(var) = UPPER(var)	Gegevens komen overeen, onafhankelijk van upper- of lowercase
ORDER BY ...	Sorteren aan de hand van een gegeven variabele
ORDER BY ... ASC	Sorteren van klein naar groot (zelfde als ORDER BY →)
ORDER BY ... DESC	Sorteren van groot naar klein
<getal>	Verwijst naar de <getal> ^{de} kolom (bv. gebruiken bij ORDER BY)
&&<variabele>	Prompt de gebruiker maar één keer om waarde in te geven, daarna kan je de waarde hergebruiken met &<variabele>
UNDEFINE <variabele>	Verwijdert de waarde die men in de variabele zette met &<var>
DEFINE <variabele>	Steekt een waarde in een variabele, zonder prompt voor gebruiker
SET VERIFY ON/OFF	Toggle of je de waarde van de gevraagde variabele weergeeft na invoer, zo kan je de waarde controleren (gebruiken vooraan script)
CASE CONVERSION	
UPPER	Zet string om naar hoofdletters
LOWER	Zet string om naar kleine letters
INITCAP	Zet enkel de eerste letter van een string als hoofdletter, de rest als kleine letters
CHARACTER MANIPULATION	
CONCAT ('Hello', 'World')	Output → 'HelloWorld'
SUBSTR ('HelloWorld', 1, 5)	Output → 'Hello'
LENGTH ('HelloWorld')	Output → 10
INSTR ('HelloWorld', 'W')	Output → 6
LPAD (salary, 10, ' *')	Left padding. Output → *****24000 → (gegeven, aantal voorziene plaatsen, aanvullen met ...)
RPAD (salary, 10, ' *')	Right padding. Output → 24000***** → (gegeven, aantal voorziene plaatsen, aanvullen met ...)
REPLACE ('string', 'A', 'B')	Vervangt, in de gegeven string, de letter A door de letter B
TRIM ('H' FROM 'HelloWorld')	Verwijdert de gegeven letter uit de gegeven string → 'elloWorld'
NUMBER FUNCTIONS	
ROUND (getal, decimalen)	Rond het getal af op de gegeven hoeveelheid decimalen
ROUND (getal, -1)	Min → rond af op gegeven positie vóór de komma (45 → 50)
TRUNC (getal, decimalen)	Snijdt het getal af op de gegeven hoeveelheid decimalen
MOD (getal, deler)	Modulo, geeft de rest van de gegeven deling als resultaat
DUAL	Dummy tabel die je kan gebruiken om resultaten van functies en bewerkingen te zien (SELECT <bewerking> FROM ... DUAL)
DATUM FUNCTIES	
SELECT SYSDATE FROM DUAL	Geeft de huidige datum weer

SYSDATE	Bevat de huidige datum
MONTHS_BETWEEN ('DD-MON-RR' , 'DD-MON-RR')	Geeft het aantal maanden tussen twee datums
ADD_MONTHS ('DD-MON-RR' , <aantal>)	Voegt kalendermaanden toe aan datum
NEXT_DAY ('DD-MON-RR' , 'FRIDAY')	De volgende gegeven dag na de datum
LAST_DAY ('DD-MON-RR')	De laatste dag van de maand in de datum
ROUND (SYSDATE , 'MONTH')	Rond de datum af op maand 25-JUL-03 → 01-AUG-03
ROUND (SYSDATE , 'YEAR')	Rond de datum af op jaar 25-JUL-03 → 01-JAN-04
TRUNC (SYSDATE , 'MONTH')	Snijdt de datum af 25-JUL-03 → 01-JUL-03
TRUNC (SYSDATE , 'YEAR')	Snijdt de datum af 25-JUL-03 → 01-JAN-03
EXPLICIT DATA CONVERSION	
TO_CHAR (date , 'format_model')	Zet het datatype om naar CHAR in het gegeven formaat → Vb. TO_CHAR (hire_date, 'fmDD Month YYYY')
TO_NUMBER (char , 'format_model')	Zet het datatype om in NUMBER
TO_DATE (char , 'format_model')	Zet datatype om naar datum → Vb. TO_DATE ('May 24, 1999' , 'Month DD, YYYY')
DATE FORMAT MODEL	
fm	Verwijdert 'padded blanks' en leidende nullen die ingebouwd zijn in sommige formaten (Bv. MONTH)
YYYY	Volledig jaar in cijfers
YEAR	Jaar volledig uitgespeld (Engels) → Hoofdletters → Heeft vast aantal karakters (zonder fm, meer witruimte)
MM	Twee cijfers voor de maand
MONTH	Volledige naam van de maand → Hoofdletters → Heeft vast aantal karakters (zonder fm, meer witruimte)
MON	3 eerste letters van de maand → Hoofdletters
DY	3 eerste letters van de dag → Hoofdletters
DAY	Volledige naam van de dag → Hoofdletters → Heeft vast aantal karakters (zonder fm, meer witruimte)
DD	Twee cijfers voor de dag
D	Één cijfer voor de dag van de week (zondag = 1) → Amerikaanse notatie: week begint op zondag
DDD	Dag in het jaar (0-365)
Month	Volledige maand, met eerste letter als hoofdletter → Heeft vast aantal karakters (zonder fm, meer witruimte)
HH24	Uur in 24-uur formaat
HH12	Uur in 12-uur formaat → Vb. 'HH12:MI:SS AM'

MI	Minuten
SS	Seconden
AM	Na tijdsformaat → geeft AM of PM weer
DD “of” MONTH	Extra strings toevoegen met "" → 12 of OCTOBER
ddspth	Output → 'fourteenth'
ddth	Output → '6th'
ddsp	Output → 'six'
Ddspth	Output → 'Fourteenth'
TO_CHAR FUNCTIES MET GETALLEN	
9	Representeert een getal Vb. TO_CHAR (salary, '\$99,999.00') → \$1,275.25
0	Geeft een 0 weer (als er niets anders weer te geven is)
\$	Plaatst een 'floating' dollarteken → floating: verplaatst mee met bv. nullen
L	Gebruikt een 'floating' teken van de lokale currency → floating: verplaatst mee met bv. nullen (achter getal)
S	Geeft een + of – teken weer
.	Print een decimaal punt
,	Print een komma tussen duizendtallen
GENERAL FUNCTIONS	
NVL (expr1, expr2)	Geeft gegeven resultaat als expr1 null is Expr1 → Wat wil je berekenen Expr2 → Wat te doen als expr1 null is Vb. NVL (job_id, 'No job yet')
NVL2 (expr1, expr2, expr3)	Zelfde als NVL, maar 3 argumenten Expr2 → Wordt getoond als expr1 <u>niet</u> null is Expr3 → Wordt getoond als expr1 null is
NULLIF (expr1, expr2)	Geeft een null-waarde terug wanneer expr1 = expr2
COALESCE (expr1, expr2, ..., exprn)	De eerste expressie die niet null is, wordt getoond → als expr1 null is en expr2 niet, wordt expr2 getoond en expr3 niet meer gecheckt
CONDITIONAL EXPRESSIONS	
CASE <kolom> WHEN ... THEN ... ELSE ... END	If-else statement (Hier kun je op andere dingen testen als =) Vb. SELECT ... CASE ... WHEN ... LIKE ... THEN ... ELSE ... END "naam kolom" FROM ...
DECODE (job_id, Search1, result1 Search2, result2, Default)	If-else statement (kan enkel op gelijkheid testen) → Als search1 = job_id → Voer result1 uit
OBTAINING DATA FROM MULTIPLE TABLES	
FROM employees JOIN departments USING (department_id)	Gaat de rijen waarin het department_id gelijk is samenvoegen
FROM <tabel1> JOIN <tabel2>	Voegt rijen samen waarin de inhoud van kolom 1 = kolom 2

ON <kolom1> = <kolom2>	3 tabellen tonen → FROM (1)... JOIN (2) ... ON ... JOIN (3)... ON ... → Tabel aliasen gebruiken om te tonen welke tabellen samenvoegen (Tabellen niet met komma's scheiden, JOIN gebruiken)
<tabel>.<kolom>	Gaat naar de gegeven kolom zoeken in de gegeven tabel → Gebruiken bij overlap van kolommen (meerdere tabellen)
FROM employees e	e is hier een alias van employees (gebruiken om tabellen te verduidelijken) → SELECT e.employee_id, e.salary
CREATING TABLES	
CREATE TABLE table (kolom datatype constraint);	Creëer een tabel met verschillende kolommen met verschillende eigenschappen
CREATE TABLE table (kolom,kolom,...) AS subquery;	Tabel creëren met een subquery
DEFAULT SYSDATE	Voeg een default waarde toe aan een kolom → Bij constraints zetten
DATA TYPES	
VARCHAR2(lengte)	String (letters, cijfers,...) met lengte van 0 tot <lengte>
CHAR(lengte)	String (letters, cijfers,...) met exact bepaalde lengte
NUMBER(totaal, decimalen)	Numerieke data → Totaal plaatsen en decimalen als parameters
DATE	Datum- en tijdswaarden
LONG	VARCHAR2, maar tot 2GB
CLOB	VARCHAR2, maar tot 4GB
RAW and LONG RAW	Raw binary data
BLOB	Binary data, tot 4GB
BFILE	Binary data vanuit een externe file, tot 4GB
ROWID	Base-64 nummer dat elke rij in een tabel een uniek adres geeft
TIMESTAMP	Datum met fractionele seconden
INTERVAL YEAR TO MONTH	Opgeslagen als interval van jaren en maanden
INTERVAL DAY TO SECOND	Opgeslagen als interval van dagen, uren, minuten en seconden
CONSTRAINTS	
NOT NULL	De kolom mag geen null-waarde bevatten
UNIQUE (<kolom>)	De waarden in deze kolom (of combinatie van kolommen) moeten uniek zijn voor alle rijen van de tabel
PRIMARY KEY (<kolom>)	Unieke identificatie van elke rij in een tabel
FOREIGN KEY <kolom> REFERENCES <tabel> (<kolom>)	Zorgt ervoor dat de waarden in de FK kolom overeen komen met de waarden in de PK kolom van de andere tabel → Makkelijker om op tabel-level of met ALTER TABLE toe te voegen
CHECK (<conditie>)	Zorgt ervoor dat de gegeven voorwaarde juist is
CONSTRAINT (naam) <constraint>	Constraints kan je een naam geven, dit is optioneel
,kolom datatype CONSTRAINT,	Kolom-level constraint, hoort enkel bij die kolom waar hij bij staat → Staat bij de kolom vóór de komma
<kolommen> ,CONSTRAINT	Tabel-level constraint, hoort bij de gespecificeerde kolom(men) → Staat na alle kolommen, ná de komma

ALTERING & DROPPING TABLES	
ALTER TABLE tabel READ ONLY;	Zorg dat anderen een table enkel kunnen lezen
ALTER TABLE tabel READ WRITE;	Zorg dat anderen de table kunnen lezen en bewerken
ALTER TABLE tabel ADD CONSTRAINT ...;	Voeg een constraint toe aan een table → Schrijven zoals constraints op tabel-level
ALTER TABLE table ADD (constr1, constr2,...);	Voeg meerdere constraints toe aan een tabel
DROP TABLE tabel;	Verwijdert een tabel
DROP TABLE tabel CASCADE CONSTRAINTS	Verwijdert een tabel en alle samenhangende constraints (PK, FK,...) → In script zetten voor CREATE TABLE
ADDING VALUES TO TABLES	
INSERT INTO tabel VALUES (value1,value2,...);	Steek waarden in een tabel → Moet direct overeenkomen met de volgorde van kolommen
INSERT INTO tabel (kolom, kolom,...) VALUES (value, value,...);	Steek warden in specifieke kolommen van een tabel → Moeten direct overeenkomen met volgorde van gegeven kolommen
UPDATE table SET kolom = value (WHERE conditie);	Verander de waarde van een gegeven in een kolom → Kan meerdere tegelijk
DELETING ROWS	
DELETE FROM TABLE tabel;	Verwijdert alle BESTAANDE rijen uit een tabel (tabel structuur blijft bestaan)
DELETE FROM table WHERE condition;	Verwijdert specifieke rijen uit een tabel
TRUNCATE TABLE tabel;	Verwijdert alle rijen uit een tabel (tabel structuur blijft bestaan) → Efficiënter dan DELETE (DDL statement), meer permanent
TRANSACTION CONTROL STATEMENTS	
COMMIT	Beëindigt de huidige transactie door alle veranderingen permanent te maken
SAVEPOINT name	Markeert een savepoint in de huidige transactie
ROLLBACK	Gaat terug naar de vorige COMMIT
ROLLBACK TO SAVEPOINT name	Gaat terug naar het gegeven SAVEPOINT
FOR UPDATE	Wacht tot een eventuele update door andere users gedaan is
GROUP FUNCTIONS	
AVG (DISTINCT/ALL n)	Het gemiddelde van n
COUNT (expr)	Telt de expr-waarde van alle rijen waar deze niet null is op → Gebruiken met *,DISTINCT of ALL
MAX (expr)	Geeft de hoogste waarde van 'expr', negeert null-waardes → Gebruiken met DISTINCT of ALL
MIN (expr)	Geeft de laagste waarde van 'expr', negeert null-waardes → Gebruiken met DISTINCT of ALL
STDDEV (n)	Berekent de standaard deviatie van n, negeert null-waardes → Gebruiken met DISTINCT of ALL
SUM (n)	Berekent de som van de waardes in 'n', negeert null-waardes → Gebruiken met DISTINCT of ALL
VARIANCE (n)	Berekent de variantie van n, negeert null-waardes

	→ Gebruiken met DISTINCT of ALL
SELECT AVG(n)	Groep functies worden achter SELECT geplaatst
GROUP BY <waarde>	Groepeert gegevens met dezelfde 'waarde'
HAVING ...	Bepaalt welke gegevens er worden weergegeven op basis van de gegeven conditie (dus een WHERE voor als je GROUP BY gebruikt)
SUBQUERIES	
WHERE salary > (<subquery>)	Manier van een subquery te gebruiken
IN (1,2,3,...)	Vergelijkt een waarde en is true als het overeenkomt met een waarde in de lijst
ANY	Vergelijkt een waarde met elke waarde in de lijst, 1 match = true → Moet een relationele operator voor staan! (=, !=, <, >, <=, >=)
ALL	Vergelijkt een waarde met elke waarde in de lijst, allemaal match = true → Moet een relationele operator voor staan! (=, !=, <, >, <=, >=)
CREATING VIEWS	
CREATE VIEW view (alias, alias,...) AS subquery... ;	Creëert een sort van read-only tabel van een SELECT statement → OR REPLACE, FORCE, NOFORCE, WITH READ ONLY,... → WITH CHECK OPTION CONSTRAINT constraint_name
DROP VIEW view;	Verwijdert een view
CREATING SEQUENCES	
CREATE SEQUENCE sequence INCREMENT BY n START WITH x MAXVALUE y NOCACHE NOCYCLE;	Creëer een sequentie (bv. Een opeenvolging van nummers)
Sequence.NEXTVAL	Geeft de volgende waarde van een sequence
Sequence.CURRVAL	Geeft de huidige waarde van een sequence
ALTER SEQUENCE sequence;	Bewerk een sequence, werkt zoals ALTER TABLE
DROP SEQUENCE sequence;	Verwijdert een sequence
CREATING INDEXES	
CREATE INDEX index ON tabel (kolom, ...);	Creëert een index op één of meerdere kolommen → Versnelt de query snelheid
DROP INDEX index;	Verwijdert een index
CREATING SYNONYMS	
CREATE SYNONYM syn FOR object;	Geef een synoniem aan een object, voor betere access
DROP SYNONYM syn;	Verwijdert een synoniem

OPERATOR	MEANING
=	Equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
<>	Not equal to
BETWEEN ... AND ...	Between two values (inclusive)
IN (set)	Match any of a list of values
LIKE	Match a character pattern (use with wildcards)
IS NULL	Is a null value
AND	Returns TRUE if both conditions are true
OR	Returns TRUE if either condition is true
NOT	Returns TRUE if condition is false