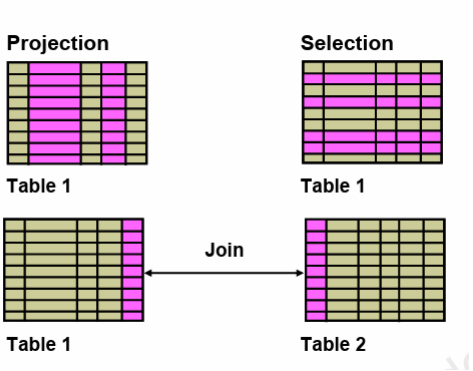
DATA SQL SAMENVATTING

SELECT Statement




```
----Basic Select Statement----
SELECT name, last_name
FROM employees;

-Alle kolommen van die tabel selecteren-
SELECT *
FROM employees;

----Plus/Min/Maal/Delen----
SELECT last_name, salary, salary + 300
FROM employees;

() gebruiken om duidelijkheid te vormen of volgorde te veranderen

----Null Values----
Een null value is geen gegevens dus geen 0, want 0 is een gegeven

----Column Aliases----
-2 manieren-
SELECT last_name AS name, commission_pct comm
FROM employees;

SELECT last_name "Name", commission_pct*12 "Annual Salary"
FROM employees;

----Concatenation----
SELECT last-name || job_id AS "Emlpoyees"
FROM employees;

----Literals----
Een nummer, Een datum, Een karakter
SELECT last_name || ' is a ' || job_id AS "Employee Details"
FROM employees;

----Comparison Operators----
= : Equel 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
IS NULL : Is a null value

SELECT last_name, salary
FROM employees
WHERE salary <= 3000;

----Duplicate Rows----
REMOVE DUPLICATE ROWS
SELECT DISTINCT department_id
FROM employees;
```

```
----Display Table Structure----
DESCRIBE employees

-DISPLAY TYPES-
NUMBER (p, s)
Number value having a maximum number of digits p, with s
digits to the right of the decimal point

VARCHAR2 ( s)
Variable-length character value of maximum size s

DATE
Date and time value between January l, 4712 B.C. and
December 31, A.D. 9999.

CHAR ( s)
Fixed-length character value of size s
```

```
----Restricting and Sorting Data----

----WHERE----
SELECT emlpoyee_id, last_name, department_id
FROM employees
-WHERE department_id = 90;
-WHERE last_name = 'Whalen'
-WHERE hire_date = '17-FEB-96' (DEFAULT DATE FORMAT: DD-MON-RR)
(Geen alias gebruiken (kan niet))

----BETWEEN----
SELECT last_name, salary
FROM employees
WHERE salary BETWEEN 2500 AND 3500;
(Tussen de 2 gegeven getallen)

----IN----
SELECT employee_id, manager_id
FROM employees
WHERE manager_id IN (100, 101, 201);
(Zit het manager id in deze lijst met items)

----LIKE----
SELECT first_name
FROM employees
WHERE first_name LIKE 'S%';
% : Geeft aan dat er geen of meer karakters mogen zijn
- : Geeft aan dat er 1 karakter mag zijn
(Karakter patroon maken)

----ESCAPE INDENTIFIER----
SELECT EMPLOYEE_ID, last_name, job_id
FROM employees
WHERE job_id LIKE '%SA\_%' ESCAPE ''\';

----NULL----
SELECT last_name, manager_id
FROM employees
-WHERE manager_id IS NULL;
-WHERE manager_id IS NOT NULL;

----LOGICAL OPERATORS AND / OR / NOT----
----AND / OR----
SELECT emlpoyee_id, last_name, job_id, salary
FROM employees
WHERE salary >= 10000
-AND job_id LIKE '%MAN%';
-OR job_id LIKE '%MAN%';
----NOT----
SELECT last_name, job_id
FROM employees
WHERE job_id NOT IN ('IT_PROG', 'ST_CLERK');

----Voorrang Tabel----
1 : Arithmetic operators
2 : Concatenation operator
3 : Comparison conditions
4 : IS [NOT] NULL, LIKE, [NOT] IN
5 : [NOT] BETWEEN
```

```
6 : Not equal to
7 : NOT logical condition
8 : AND logical condition
9 : OR logical condition
POWERMOVE : ()

----ORDER BY----
-ASC : Ascending order, default
-DESC : Descending order
SELECT last_name, job_id, department_id, hire_date
FROM employees
-ORDER BY hire_date
-ORDER BY hire_date DESC;
-ORDER BY annsal (Dit is een aslias dat is dul mogelijk as dit bij de select is gegeven
-ORDER BY 3; (De 3 is de plaats waar de colom staat in de SELECT)
-ORDER BY department_id, salary DESC; (Meerdere kolommen)

----Variabelen / Input----
&
SELECT employee_id, last_name, salary, department_id
FROM employees
-WHERE employee_id = &employee_num;
-WHERE job_id = '&job_title' (Plaats het in elke quotes anders moet je ze zelf invoeren);

&& (Gebruik je wanneer je niet steeds om de zelfde variabelen wilt vragen)
SELECT employee_id, last_name, job_id, &&column_name
FROM employees
ORDER BY &column_name;

-----DEFINE / UNDEFINE---- (Variabelen die je hebt ingevuld verwijderen)
DEFINE employee_num = 200

SELECT employee_id, last_name, salary, department_id
FROM employees
WHERE employee_id = &employee_num;

UNDEFINE employee_num

----VERIFY----
(Dit laat zien wat SQL+ doet
SET VERIFY ON
```

```
----Hoofdstuk 3----
----Using Single-Row Functions to Customize Ourput-----
-----Case-Conversion Functions-----
SELECT employee_id, last_name, department_id
FROM employees
WHERE LOWER(last_name) = 'higgins';
- WHERE UPPER(last_name = 'HIGGINS'
- WHERE INITCAP(last_name) = 'Higgins'

-----Character-Manipulation Functions-----
CONCAT('Hello', 'World') --> HelloWorld
SUBSTR('HelloWord',1,5) --> Hello
LENGTH('HelloWorld') --> 10
INSTR('HelloWorld', 'W') --> 6
LPAD(salary, 10, '*') --> *****24000
RPAD(salary, 10, '*') --> 24000*****
REPLACE('JACK and JUE', 'J', 'BL') --> BLACK and BLUE
TRIM('H' FROM 'HelloWorld') --> elloWorld
--
SELECT employee_id, CONTACT(first_name, last_name) NAME, job_id, LENGTH (last_name),
INSTR(last_name, 'a') "Contains 'a'?"
FROM employees
WHERE SUBSTR(job_id, 4) = 'REP';

-----Number Functions-----
ROUND(45.926, 2) --> 45.93
TRUNC(45.926, 2) --> 45.92
MOD(1600, 300) --> 100
---Round
SELECT ROUND(45.923, 2), ROUND(45.923, 0), ROUND(45.923, -1)
FROM DUAL;

-----DUAL-----
DUAL is a dummy table that you can use to view results from functions and calculations
```

```
-----DATES-----
---DEFAULT DISPLAY FORMAT
---DD-MON-RR

SELECT last_name, hire_date
FROM employees
WHERE hire_date < '01-FEB-88';

---SYSDATE---(Datum van de server / tijd van de server)
SELECT sysdate
FROM dual;

---CURRENT_DATE---(Datum van de sessie op je computer / tijd ook)
SELECT current_date
FROM dual

---Arithmetic with Dates---
data + number --> Adds days (Date Ouput)
date - number --> remove days (Date Output)
date - date --> Subrtracts one date from the other (Number of days Output)
date + number/24 --> Add a number of hours to a date (Date Ouput)
VB:
SELECT last_name, (SYSDATE-hire_date)/7 AS WEEKS
FROM employees
WHERE department_id = 90;

-----Date-Manipulation Functions-----(173)
MONTHS_BETWEEN ('01-SEP-95','11-JAN-94') --> 19.6774194
--> Number of months between two dates
ADD_MONTHS ('32-JAN-96', 1) --> '29-FEB-96'
 --> Add calendar months to date
NEXT_DAY ('01-SEP-95','FRIDAY') --> '08-SEP-95'
 --> Next day of the date specified
LAST_DAY ('01-FEB-95') --> '28-FEB-95'
 --> Last day of the month
VB:
SELECT employee_id, hire_date, MONTHS_BETWEEN (SYSDATE, hire_date) TENURE, ADD_MONTHS (hire_date 6) REVIEW, NEXT_DAY (hire_d
FROM employees
WHERE MONTHS_BETWEEN (SYSDATE, hire_date) < 100;

--ROUND AND TRUNC WITH DATES
ROUND --> Round date
TRUNC --> Truncate date
-
-
-
- Vraag Hoe dit werkt
-
-
-
-
-
```

```
-----Hoofdstuk 4-----

-----Explicit Data Type Conversion-----

---CHARACTER --> NUMBER---
TO_NUMBER()
---NUMBER --> CHARACTER---
TO_CHAR(number, 'format_model')
SELECT TO_CHAR(salary, '$99,999.00') SALARY
FROM employees
WHERE last_name = 'Ernst';

---CHARACTER --> DATE---
TO_DATE()
---DATE --> CHARACTER---
TO_CHAR(date, 'format_model')
SELECT employee_id, TO_CHAR(hire_date, 'MM/YY') Month_Hired
FROM employees
WHERE last_name = 'Higgins';

-----DATE FORMAT MODEL-----(195) DATUM
YYYY --> Full year in numbers
YEAR --> Year spelled out (in English)
MM --> Two-digit value for the month
```

```
MONTH --> Full name of the month
MON --> Three-letter abbreviation of the month
DY --> Three-letter abbreviation of the day of the week
DAY --> Full name of the day of the week
DD --> Numeric day of the month

-----DATE FORMAT MODEL-----(179) TIJD
-----To remove padded blanks or to suppress leading zeros, use the fill mode fmelement.-----

-----NVL Function-----
--> Converts a null value to an actual value
SELECT last_name, salary, NVL(commission_pct, 0), (salary*12) + ((salary*12) + (salary*12*NVL(commission_pct, 0)) AN_SAL
FROM employees;

-----NVL2 Function-----(213)
--> If expr1 is not null, NVL2 returns expr2. If expr1 is null, NVL2 returns expr3. The argument expr1 can have any data typ
SELECT last_name, salary, commission_pct, NVL2(commission_pct, 'SAL+COMM', 'SAL') income
FROM employees;
-----NULLIF-----(214)
--> Compares two expressions and returns null if they are equal; returns the first expression if the are not equa)l
SELECT first_name, LENGTH(first_name "expr1", last_name, LENGHT(last_name) "expr2", NULLIF (LENGTH(first_name), LENGTH(last_
FROM employees;

-----COALESCE Function----
--> Returns the first non-null ex ression in the expression list
SELECT last_name, employee_id,
COALESCE(TO_CHAR(commission_pct),TO_CHAR(manager_id), 'No commission and no manager')
```

```
-----Hoofdstuk 5-----(232)
-----Group Functions-----
AVG() --> Gemiddelde
AVG(NVL(commission_pct, 0)) --> Gemiddelde (NVL vervangt de nullwaardes naar een 0
COUNT() --> Tellen
COUNT(DISTINCT department_id) --> Tellen
MAX() --> Maximaal
MIN() --> Minimaal
STDDEV()
SUM() --> Som
VARIANCE()
SELECT group_function(coloumn), ...
FROM table
WHERE [condition]
ORDER BY [column]

-----GROUP BY-----(245)
SELECT department_id, AVG(salary)
FROM employees
GROUP BY department_id;

-----GROUP BY (GROUPS WITHIN GROUPS)-----
SELECT department_id, job_id, sum(salary)
FROM employees
GROUP BY department_id, job_id
ORDER BY job_id;

-----Restricting Group Results-----(252)
-----HAVING-----(Hetzelfde als where maar voor groepsfuncties)
SELECT job_id, SUM(salary) PAYROLL
FROM employees
WHERE job°id NOT LIKE '%REP%'
GROUP BY job_id
HAVING SUM(salary) > 13000
ORDER BY SUM(salary);

-----Nesting Group Functions-----(257)
SELECT MAX(AVG(salary))
FROM employees
GROUP BY department_id;
--> Dit is niet duidelijk je kunt niet zien welk department dit is
SELECT department_id, MAX(AVG(salary))
FROM employees
GROUP BY department_id;
```

```
-----Hoofdstuk 6-----
-----Displaying Data from Multiple Tables-----
---USING---
```

```
SELECT employee_id, last_name, location_id, department_id
FROM employees JOIN departments
USING (department_id)

---ON---
SELECT e.employee_id, e.last_name, e.department_id, d.department_id, d.location_id
FROM employees e JOIN departments d
ON (e.department_id = d.department_id);

---Three-Way Joins ON---
SELECT employee_id, city, department_name
FROM employees e
JOIN departments d
ON d.department_id = e.department_id
JOIN locations l
ON d.location_id = l.location_id;

---AND / WHERE BY THE ON CLAUSE---
SELECT e.employee_id, e.last_name, e.department_id, d.department_id, d.location_id
FROM employees e JOIN departments d
ON (e.department_id = d.department_id)
AND e.manger id = 149;

SELECT e.employee_id, e.last_name, e.department_id, d.department_id, d.location_id
FROM employee e JOIN departments d
ON (e.department_id = d.department_id)
WHERE e.manager_id = 149;

---SELF-JOIN USING ON CLAUSE---
SELECT worker.last_name emp, manager.last_name mgr
FROM employees worker JOIN employees manager
ON (worker.manager_id = manager.employee_id);


---Nonequijoins---(287)
SELECT e.last_name, e.salary, J;grade_level
FROM employees e JOIN job_grades j
ON e.salary BETWEEN j.lowest_sal AND j.highest_sal;

-----OUTER JOINS-----(289)
(TO return the department record that does not havr any employees, or employees that do not have an assigned departmnet, you

---LEFT OUTER JOIN---
SELECT e.last_name, e.department_id, d.department_name
FROM employees e LEFT OUTER JOIN departments d
ON (e.department_Id = d.department_id);

---RIGHT OUTER JOIN---
SELECT e.last_name, e.department_id, d.department_name
FROM employees e RIGHT OUTER JOIN departments d
ON (e.department_Id = d.department_id);

---FULL OUTER JOIN---
SELECT e.last_name, e.department_id, d.department_name
FROM employees e FULL OUTER JOIN departments d
ON (e.department_Id = d.department_id);

---CROSS JOINS (Cartesian Product)---
SELECT last_name, department_name
FROM employees
CROSS JOIN departments;
```

Vragen pagina 279

```
-----Hoofdstuk 7-----
---Single-Row Subqueries---
= --> Equal to
> --> Greater than
>= --> Greater than or equal to
< --> Less than
<= --> Less than or equal to
<> --> Not equal to

SELECT last_name, job_id, salary
FROM employees
WHERE job_id = (SELECT job_id FROM employees WHERE last_name = 'Taylor')
AND salary > (SELECT salary FROM employees WHERE last_name = 'Taylor');
```

```
---HAVING WITH SUBQUERIES---
SELECT department_id, MIN(salary)
FROM employees
GROUP BY department_id
HAVING MIN(salary) > (SELECT MIN(salary) FROM employees WHERE department_id = 50);

---Multiple-Row Subqueries---(322)
IN --> Equal to any member in the list
ANY --> Must be preceded by =, !=, >, <, <=, >=.
                Compares a value to each value in a list or
                returned by a query. Evaluates to FALSEif the
                query returns no rows.
ALL --> Must be preceded by =, !=, >, <, <=, >=.
                Compares a value to every value in a list or
                returned by a query. Evaluates to TRUEif the
                query returns no rows.

SELECT employee_id, last_name, job_id, salary
FROM employees
WHERE salary < ANY (SELECT salary FROM employees WHERE job_id = 'IT_PROG')
AND job_id <> 'IT_PROG';

---Null Values---(326)
```

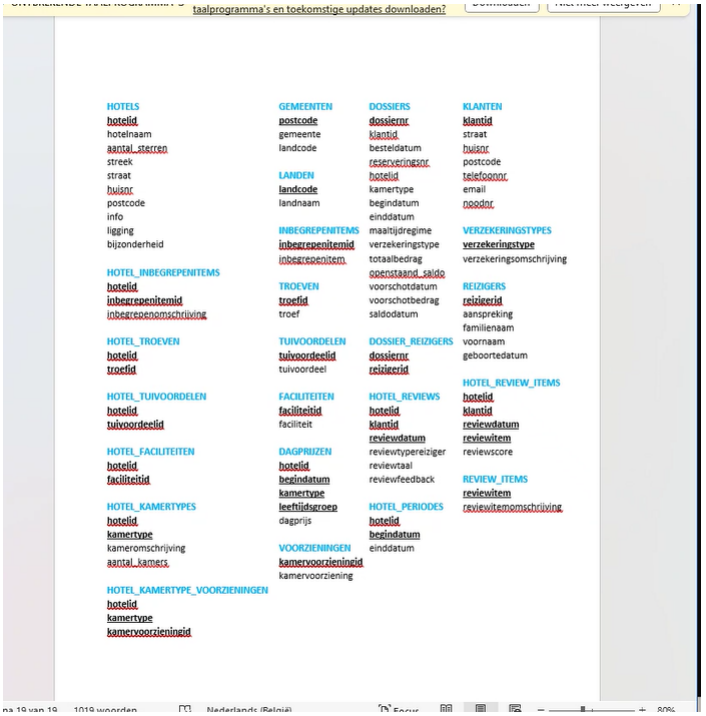
```
-----Hoofdstuk 9-----
---INSERT DATA---
INSERT INTO departments(department_id, department_name, manager_id, location_id)
VALUES (70, 'Public Relations', 100, 1700);

---INSERT DATA WITH NULL VALUES---(341) Explicit Implicit
INSERT INTO departments
VALUES (100, 'Finance', NULL, NULL);

---Inserting Specific Date and Time Values---
INSERT INTO employees
VALUES (114, 'DEN', 'RAPhealy', 'DRAPHEAL', '515.127.4561', TOD_DATE('FEB 3, 1999', 'MON DD, YYYY'), 'SA_REP', 11000, 0.2, 1

---Copying ROws from Another Table(345)---

-----Updating Rows in a Table-----(349)
UPDATE employees
SET department_id = 50
WHERE employee_id = 113;

---Upadating Two Columns with a Subquerry---
UPDATE employees
SET job_id = (SELECT job_id FROM employees WHERE employee_id = 205), salary = (SELECT salary FROM employees WHERE employee_i
WHERE employee_id = 113

-----Delete Rows from a Table-----
DELETE FROM departments
WHERE department_name = 'Finance';

---Delete Rows BAed on Another Table---
DELETE FROM employees
WHERE department-id = (SELECT department_id FROM departments WHERE department_name LIKE '%Public%');

-----TRUNCATE-----
Verwijder alle data uit een tabel
TRUNCATE TABLE employees;
```

```
-----Hoofdstuk 10-----
-----Create Table-----
CREATE TABLE dept (
deptno NUMBER(2),
[Constraint Column Level]
dname VARCHAR2(14),
loc VARCHAR2(13),
create_date DATE DEFAULT SYSDATE);

[Constraint Table Level]

-----CONFIRM TABLE CREATION-----
DESCRIBE dept
```

```
---DATA TYPES(379)---

-----CONTRAINTS-----
---COLUMN-LEVEL---
CREATE TABLE employees(
employee_id NUMBER(6) CONSTRAINT emp_emp_id_pk ¨PRIMARY KEY,
--employee_id NUMBER(6) CONSTRAINT emp_salary_min CHECK (salary > 0),
first_name VARCHAR2(20),
....);

---TABLE-LEVEL---
CREATE TABLE employees(
employee_id NUMBER(6),
first_name VARCHAR2(20),
...
job_id VARCHAR2(10) NOT NULL,
CONSTRAINT emp_emp_id_pk
    PRIMARY KEY (EMPLOYEE_ID));

-----CONSTRAINT NAMING-----
NOT NULL
UNIQUE
PRIMARY KEY
FOREIGN KEY
(Voeg hier benamingsregels toe)

-----Creating a Table Using a Subquery-----(418)

-----Dropping a Table-----
DROP TABLE dept80;
DROP TABLE dept80 casscade constraints;
```

```
-----Hoofdstuk 11-----
-----Create a View----
CREATE OR REPLACE empvu80
AS SELECT employee_id, last_name, salary
FROM employees
WHERE department_id = 80;
--WITH CHECK OPTION CONSTRAINT empvu20_ck; --> Hierbij blijf je binnen in het view en kun je er buiten niks aanpassen

-----Checking if a view is correct made-----
DESCRIBE empvu80

-----SEQUENCES-----
-----CREATE A SEQUENCE-----(455)
CREATE SEQUENCE dept_deptid_seq
--Voor meer opties hieronder zie cursus--

-----Using a Sequence-----
INSERT INTP departments(
department_id, departments_name, location_id)
VALUES (dept_deptid_seq.NEXTVAL, 'Support', 2500);

458
```

EEN VRAAG WAAR JE 5 TABELLEN MOET



Op voorhand bepalen

FM? → haalt alle nullen weg die niet nodig zij en haalt alle nullen weg