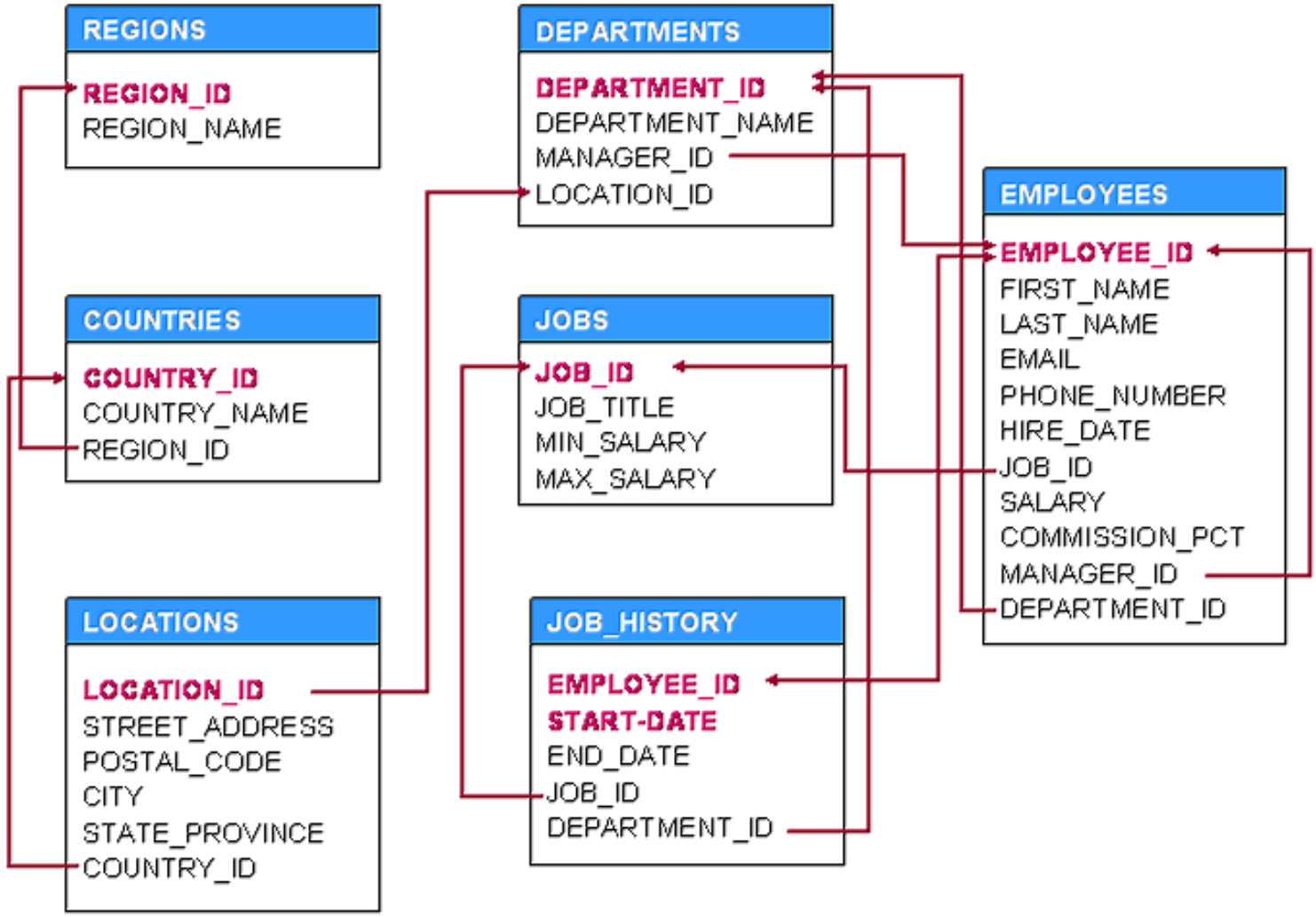
TECNICAS DE ALMACENAMIENTO DE DATOS

DER – Esquema HR



Consultas Simples

```
--TODOS LOS CAMPOS
--TODOS LOS REGISTROS
SELECT *
  FROM TABLA;
```

```
--SOLO 2 CAMPOS
--TODOS LOS REGISTROS
SELECT CAMPO1, CAMPO2
  FROM TABLA;
```

```
--TODOS LOS CAMPOS
--FILTRO REGISTROS
SELECT *
  FROM TABLA
 WHERE CAMPO = VALOR;
```

```
--SOLO 2 CAMPOS
--FILTRO REGISTROS
--ORDENADO
SELECT CAMPO1, CAMPO2
  FROM TABLA
 WHERE CAMPO IS NULL
 ORDER BY CAMPO1;
```

Consultas Simples

```
--TODOS LOS CAMPOS
--TODOS LOS REGISTROS
SELECT *
  FROM EMPLOYEES;
```

```
--SOLO 2 CAMPOS
--TODOS LOS REGISTROS
SELECT FIRST_NAME,
       LAST_NAME
  FROM EMPLOYEES;
```

```
--TODOS LOS CAMPOS
--FILTRO REGISTROS
SELECT *
  FROM EMPLOYEES
 WHERE EMPLOYEE_ID = 100;
```

```
--SOLO 2 CAMPOS
--FILTRO REGISTROS
--ORDENADO
SELECT EMPLOYEE_ID, HIRE_DATE
  FROM EMPLOYEES
 WHERE DEPARTMENT_ID IS NULL
 ORDER BY EMPLOYEE_ID;
```

Consultas Simples

```
--CONCATENACION + ORDENAMIENTO
SELECT EMPLOYEE_ID,
       LAST_NAME || ', ' || FIRST_NAME AS APELLIDO_NOMBRE,
       EMAIL || '@DOMINIO.COM' AS MAIL_LABORAL
  FROM EMPLOYEES
 WHERE DEPARTMENT_ID IS NOT NULL
 ORDER BY EMPLOYEE_ID;
```

EMPLOYEE_ID	APELLIDO_NOMBRE	MAIL_LABORAL
100	King, Steven	SKING@DOMINIO.COM
101	Kochhar, Neena	NKOCHHAR@DOMINIO.COM
102	De Haan, Lex	LDEHAAN@DOMINIO.COM
103	Hunold, Alexander	AHUNOLD@DOMINIO.COM
104	Ernst, Bruce	BERNST@DOMINIO.COM
105	Austin, David	DAUSTIN@DOMINIO.COM
106	Pataballa, Valli	VPATABAL@DOMINIO.COM
107	Lorentz, Diana	DLORENTZ@DOMINIO.COM
108	Greenberg, Nancy	NGREENBE@DOMINIO.COM
109	Faviet, Daniel	DFAVIET@DOMINIO.COM
110	Chen, John	JCHEN@DOMINIO.COM

FUNCIONES UTILES

- NVL
- TO_CHAR
- TO_DATE
- TO_NUMBER
- SUBSTR
- TRIM
- UPPER / LOWER / INITCAP

http://www.techonthenet.com/oracle/functions/

Consultas Simples - NVL

```
--EJEMPLO
SELECT LOCATION_ID,
       CITY,
       NVL(STATE_PROVINCE,'DESCONOCIDA') AS PROVINCIA
  FROM LOCATIONS;
```

LOCATION_ID	CITY	PROVINCIA
1000	Roma	DESCONOCIDA
1100	Venice	DESCONOCIDA
1200	Tokyo	Tokyo Prefecture
1300	Hiroshima	DESCONOCIDA
1400	Southlake	Texas
1500	South San Francisco	California
1600	South Brunswick	New Jersey
1700	Seattle	Washington
1800	Toronto	Ontario
1900	Whitehorse	Yukon
2000	Beijing	DESCONOCIDA

Consultas Simples – TO_CHAR

```
--EJEMPLO
SELECT EMPLOYEE_ID,
       SALARY,
       TO_CHAR(SALARY,'$99,999.00') AS SALARIO1,
       TO_CHAR(SALARY,'$9,999.00') AS SALARIO2,
       HIRE_DATE,
       TO_CHAR(HIRE_DATE,'MONTH') AS MES_TXT,
       TO_CHAR(HIRE_DATE,'DDD') AS DIA_DEL_AÑO_TXT
  FROM EMPLOYEES
 WHERE TO_CHAR(HIRE_DATE,'DDD') < 100
   AND SALARY > 3000;
```

EMPLOYEE_ID	SALARY	SALARIO1	SALARIO2	HIRE_DATE	MES_TXT	DIA_DEL_AÑO_TXT
102	17000	$17,000.00	##########	13/01/2001	ENERO	013
103	9000	$9,000.00	$9,000.00	03/01/2006	ENERO	003
106	4800	$4,800.00	$4,800.00	05/02/2006	FEBRERO	036
107	4200	$4,200.00	$4,200.00	07/02/2007	FEBRERO	038
112	7800	$7,800.00	$7,800.00	07/03/2006	MARZO	066
142	3100	$3,100.00	$3,100.00	29/01/2005	ENERO	029
146	13500	$13,500.00	##########	05/01/2005	ENERO	005
147	12000	$12,000.00	##########	10/03/2005	MARZO	069
149	10500	$10,500.00	##########	29/01/2008	ENERO	029

Consultas Simples – TO_DATE

```
--EJEMPLO
SELECT TO_DATE('20160101','YYYYMMDD') AS FECHA1,
       TO_DATE('01012016','DDMMYYYY') AS FECHA2,
       TO_DATE('2016/01/01','YYYY/MM/DD') AS FECHA3,
       TO_DATE('01/01/2016','DD/MM/YYYY') AS FECHA4,
       TO_DATE('01/01/2016 08:15:25',
               'DD/MM/YYYY HH:MI:SS') AS FECHA5
  FROM DUAL;
```

FECHA1	FECHA2	FECHA3	FECHA4	FECHA5
01/01/2016	01/01/2016	01/01/2016	01/01/2016	01/01/2016

Consultas Simples – TO_NUMBER

```
--EJEMPLO
SELECT CITY,
       POSTAL_CODE,
       TO_NUMBER(POSTAL_CODE) CP_TXT
  FROM LOCATIONS
 WHERE CITY LIKE 'B%'
    OR CITY LIKE 'R%';
```

CITY	POSTAL_CODE	CP_TXT
Roma	00989	989
Beijing	190518	190518
Bombay	490231	490231
Bern	3095	3095

Consultas Simples

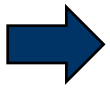
```
-- UPPER / LOWER / INITCAP
SELECT STREET_ADDRESS,
       UPPER(STREET_ADDRESS) DOMICILIO1,
       LOWER(STREET_ADDRESS) DOMICILIO2,
       INITCAP(STREET_ADDRESS) DOMICILIO3
  FROM LOCATIONS
 ORDER BY LOCATION_ID DESC ;
```

STREET_ADDRESS	DOMICILIO1	DOMICILIO2	DOMICILIO3
Mariano Escobedo 9991	MARIANO ESCOBEDO 9991	mariano escobedo 9991	Mariano Escobedo 9991
Pieter Breughelstraat 837	PIETER BREUGHELSTRAAT 837	pieter breughelstraat 837	Pieter Breughelstraat 837
Murtenstrasse 921	MURTENSTRASSE 921	murtenstrasse 921	Murtenstrasse 921
20 Rue des Corps-Saints	20 RUE DES CORPS-SAINTS	20 rue des corps-saints	20 Rue Des Corps-Saints
Rua Frei Caneca 1360	RUA FREI CANECA 1360	rua frei caneca 1360	Rua Frei Caneca 1360
Schwanthalerstr. 7031	SCHWANTHALERSTR. 7031	schwanthalerstr. 7031	Schwanthalerstr. 7031
9702 Chester Road	9702 CHESTER ROAD	9702 chester road	9702 Chester Road
Magdalen Centre, The O	MAGDALEN CENTRE, THE OX	magdalen centre, the o	Magdalen Centre, The O

Consultas Simples - DISTINCT

- La clausula DISTINCT elimina duplicados del resultado de una consulta

```
-- DISTINCT
SELECT DISTINCT REGION_ID
  FROM COUNTRIES
 ORDER BY 1;
```



REGION_ID
1
2
3
4
(null)

```
SELECT *
  FROM REGIONS;
```

REGION_ID	REGION_NAME
1	Europe
2	Americas
3	Asia
4	Middle East and Africa
999	Antartida

```
SELECT *
  FROM COUNTRIES;
```

COUNTRY_ID	COUNTRY_NAME	REGION_ID
AL	Albania	(null)
AR	Argentina	2
AU	Australia	3
BE	Belgium	1
BR	Brazil	2
CA	Canada	2
CH	Switzerland	1
CN	China	3
DE	Germany	1
DK	Denmark	1
EG	Egypt	4

Ejercicio - DISTINCT

- Elaborar una sentencia SQL que muestre los distintos años en los que se contrató personal.
  Ordenar el resultado por año.

```
SELECT DISTINCT CAMPO
  FROM TABLA;
```

Ejercicio - DISTINCT

- Elaborar una sentencia SQL que muestre los distintos años en los que se contrató personal.
  Ordenar el resultado por año.

```
-- DISTINCT
SELECT DISTINCT TO_CHAR(HIRE_DATE,'YYYY') AS AÑOS
  FROM EMPLOYEES
 ORDER BY AÑOS;
```

AÑOS
2001
2002
2003
2004
2005
2006
2007
2008

Consultas Sumarias - COUNT

- La función COUNT devuelve el resultado de contabilizar una expresión

```
SELECT COUNT(*) AS CANTIDAD_EMPLEADOS
  FROM EMPLOYEES;
```

CANTIDAD_EMPLEADOS
107

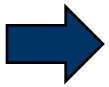
```
--CON DISTINCT
SELECT COUNT(JOB_ID) AS CANT_FUNCIONES1,
       COUNT(DISTINCT JOB_ID) AS CANT_FUNCIONES2
  FROM EMPLOYEES;
```

CANT_FUNCIONES1	CANT_FUNCIONES2
107	19

Consultas Sumarias - COUNT

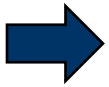
- La función COUNT devuelve el resultado de contabilizar una expresión

```
SELECT COUNT(*)
  FROM EMPLOYEES
 WHERE DEPARTMENT_ID = 50;
```



COUNT(*)
45

```
SELECT COUNT(*)
  FROM EMPLOYEES
 WHERE SALARY BETWEEN 10000
                  AND 20000;
```



COUNT(*)
18

Ejercicio - COUNT

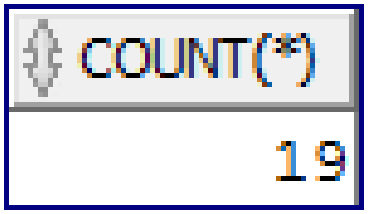
- Elaborar una sentencia SQL que permita obtener la cantidad de empleados que ingresaron en el año 2007 (tabla EMPLOYEES)

```
SELECT COUNT(*)
  FROM TABLA;
```

Ejercicio - COUNT

- Elaborar una sentencia SQL que permita obtener la cantidad de empleados que ingresaron en el año 2007

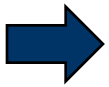
```
SELECT COUNT(*)
  FROM EMPLOYEES
 WHERE TO_CHAR(HIRE_DATE,'YYYY') = '2007';
```



Consultas Sumarias - SUM

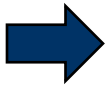
- La función SUM devuelve el resultado de sumar una expresión

```
--TOTAL SUELDOS DEPTO 50
SELECT SUM(SALARY)
  FROM EMPLOYEES
 WHERE DEPARTMENT_ID = 50;
```



SUM(SALARY)
156400

```
--TOTAL SUELDOS PROGRAMADORES
SELECT SUM(SALARY)
  FROM EMPLOYEES
 WHERE JOB_ID = 'IT_PROG';
```



SUM(SALARY)
28800

Consultas Sumarias – MIN / MAX

- La función MIN devuelve el valor mínimo de una expresión
- La función MAX devuelve el valor máximo de una expresión

```
--SALARIOS MINIMO Y MAXIMO
SELECT MIN(SALARY) AS MENOR_SALARIO,
       MAX(SALARY) AS MAYOR_SALARIO
  FROM EMPLOYEES;
```

MENOR_SALARIO	MAYOR_SALARIO
2100	24000

Ejercicio – MIN / MAX

- Elaborar una sentencia SQL que muestre los sueldos mínimo y máximo de los empleados que pertenecen al departamento de Finanzas (DEPARTMENT_ID = 100)

```
SELECT MIN(CAMPO)
  FROM TABLA;

SELECT MAX(CAMPO)
  FROM TABLA;
```

Ejercicio – MIN / MAX

- Elaborar una sentencia SQL que muestre los sueldos mínimo y máximo de los empleados que pertenecen al departamento de Finanzas (DEPARTMENT_ID = 100)

```
--SALARIOS MINIMO Y MAXIMO
SELECT MIN(SALARY) AS MENOR_SALARIO,
       MAX(SALARY) AS MAYOR_SALARIO
  FROM EMPLOYEES
 WHERE DEPARTMENT_ID = 100;
```

MENOR_SALARIO	MAYOR_SALARIO
6900	12008

Consultas Sumarias – GROUP BY

- La clausula GROUP BY se usa en las sentencias SELECT para agrupar múltiples registros mediante una o mas columnas y aplicarle una función sumaria

```
--SET DE DATOS A AGRUPAR
SELECT DEPARTMENT_ID, SALARY
  FROM EMPLOYEES
 ORDER BY 1;
```

DEPARTMENT_ID	SALARY
10	4400
20	13000
20	6000
30	11000
30	3100
30	2900
30	2800
30	2600
30	2500
40	6500
50	8000

```
--TOTALES x DEPTO
SELECT DEPARTMENT_ID,
       SUM(SALARY)
  FROM EMPLOYEES
 GROUP BY DEPARTMENT_ID
 ORDER BY 1 DESC;
```

DEPARTMENT_ID	SUM(SALARY)
10	4400
20	19000
30	24900
40	6500
50	156400
60	28800
70	10000
80	304500
90	58000
100	51608
110	20308
(null)	7000

Consultas Sumarias – GROUP BY

```
--SALARIOS TOTALES x DEPTO Y FUNCION
SELECT DEPARTMENT_ID, JOB_ID, SUM(SALARY)
  FROM EMPLOYEES
 GROUP BY DEPARTMENT_ID, JOB_ID
 ORDER BY 1, 2;
```

--SET DE DATOS A AGRUPAR

DEPARTMENT_ID	JOB_ID	SALARY
10	AD_ASST	4400
20	MK_MAN	13000
20	MK_REP	6000
30	PU_CLERK	2900
30	PU_CLERK	3100
30	PU_CLERK	2500
30	PU_CLERK	2800
30	PU_CLERK	2600
30	PU_MAN	11000
40	HR_REP	6500
50	SH_CLERK	3900

DEPARTMENT_ID	JOB_ID	SUM(SALARY)
10	AD_ASST	4400
20	MK_MAN	13000
20	MK_REP	6000
30	PU_CLERK	13900
30	PU_MAN	11000
40	HR_REP	6500
50	SH_CLERK	64300
50	ST_CLERK	55700
50	ST_MAN	36400
60	IT_PROG	28800
70	PR_REP	10000

Consultas Sumarias – GROUP BY

```
--SALARIOS TOTALES Y CANT EMPLEADOS x DEPTO Y FUNCION
SELECT DEPARTMENT_ID, JOB_ID,
       COUNT(EMPLOYEE_ID),
       SUM(SALARY)
  FROM EMPLOYEES
 GROUP BY DEPARTMENT_ID, JOB_ID
 ORDER BY 1, 2;
```

DEPARTMENT_ID	JOB_ID	COUNT(EMPLOYEE_ID)	SUM(SALARY)
10	AD_ASST	1	4400
20	MK_MAN	1	13000
20	MK_REP	1	6000
30	PU_CLERK	5	13900
30	PU_MAN	1	11000
40	HR_REP	1	6500
50	SH_CLERK	20	64300
50	ST_CLERK	20	55700
50	ST_MAN	5	36400
60	IT_PROG	5	28800
70	PR_REP	1	10000

Consultas Sumarias – HAVING

- La clausula HAVING se usa en combinación con la clausula GROUP BY para restringir los grupos de registros retornados a aquellos que cumplan la condición establecida

```
--SALARIOS TOTALES Y CANT EMPLEADOS x DEPTO Y FUNCION
--PARA DEPTOS CON MAS DE 5 EMPLEADOS
SELECT DEPARTMENT_ID, JOB_ID,
       COUNT(EMPLOYEE_ID),
       SUM(SALARY)
  FROM EMPLOYEES
 GROUP BY DEPARTMENT_ID,
       JOB_ID
HAVING COUNT(EMPLOYEE_ID) > 4
 ORDER BY 1, 2;
```

DEPARTMENT_ID	JOB_ID	COUNT(EMPLOYEE_ID)	SUM(SALARY)
30	PU_CLERK	5	13900
50	SH_CLERK	20	64300
50	ST_CLERK	20	55700
50	ST_MAN	5	36400
60	IT_PROG	5	28800
80	SA_MAN	5	61000
80	SA_REP	29	243500
100	FI_ACCOUNT	5	39600

Consultas Sumarias – HAVING

- La clausula HAVING se usa en combinación con la clausula GROUP BY para restringir los grupos de registros retornados a aquellos que cumplan la condición establecida

```
--SALARIOS TOTALES x DEPTO Y FUNCION
--PARA DEPTOS CON MAS DE 5 EMPLEADOS
SELECT DEPARTMENT_ID, JOB_ID,
       SUM(SALARY)
  FROM EMPLOYEES
 GROUP BY DEPARTMENT_ID, JOB_ID
HAVING COUNT(EMPLOYEE_ID) > 4
 ORDER BY 1, 2;
```

DEPARTMENT_ID	JOB_ID	SUM(SALARY)
30	PU_CLERK	13900
50	SH_CLERK	64300
50	ST_CLERK	55700
50	ST_MAN	36400
60	IT_PROG	28800
80	SA_MAN	61000
80	SA_REP	243500
100	FI_ACCOUNT	39600

Subconsultas

- Son consultas dentro de otras consultas
- Pueden estar en las clausulas WHERE, FROM y SELECT

```
--SUBQUERY EN WHERE
SELECT E.EMPLOYEE_ID, E.FIRST_NAME, E.LAST_NAME, E.HIRE_DATE, E.JOB_ID
  FROM EMPLOYEES E
 WHERE E.DEPARTMENT_ID = ( SELECT DEPARTMENT_ID
                             FROM DEPARTMENTS
                            WHERE DEPARTMENT_NAME = 'Finance' );
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	HIRE_DATE	JOB_ID
108	Nancy	Greenberg	17/08/2002	FI_MGR
109	Daniel	Faviet	16/08/2002	FI_ACCOUNT
110	John	Chen	28/09/2005	FI_ACCOUNT
111	Ismael	Sciarra	30/09/2005	FI_ACCOUNT
112	Jose Manuel	Urman	07/03/2006	FI_ACCOUNT
113	Luis	Popp	07/12/2007	FI_ACCOUNT

Subconsultas

- Son consultas dentro de otras consultas
- Pueden estar en las clausulas WHERE, FROM y SELECT

```
--SUBQUERY EN FROM
SELECT E.EMPLOYEE_ID, E.FIRST_NAME, E.LAST_NAME, E.HIRE_DATE, E.JOB_ID
  FROM EMPLOYEES E,
       ( SELECT DEPARTMENT_ID
           FROM DEPARTMENTS
          WHERE DEPARTMENT_NAME = 'Finance' ) D
 WHERE E.DEPARTMENT_ID = D.DEPARTMENT_ID ;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	HIRE_DATE	JOB_ID
108	Nancy	Greenberg	17/08/2002	FI_MGR
109	Daniel	Faviet	16/08/2002	FI_ACCOUNT
110	John	Chen	28/09/2005	FI_ACCOUNT
111	Ismael	Sciarra	30/09/2005	FI_ACCOUNT
112	Jose Manuel	Urman	07/03/2006	FI_ACCOUNT
113	Luis	Popp	07/12/2007	FI_ACCOUNT

Subconsultas

- Son consultas dentro de otras consultas
- Pueden estar en las clausulas WHERE, FROM y SELECT

```
--SUBQUERY EN SELECT
SELECT E.EMPLOYEE_ID, E.FIRST_NAME, E.LAST_NAME, E.HIRE_DATE, E.JOB_ID,
       ( SELECT DEPARTMENT_NAME
           FROM DEPARTMENTS D
          WHERE D.DEPARTMENT_ID = E.DEPARTMENT_ID ) AS DEPARTAMENTO
  FROM EMPLOYEES E;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	HIRE_DATE	JOB_ID	DEPARTAMENTO
100	Steven	King	17/06/2003	AD_PRES	Executive
101	Neena	Kochhar	21/09/2005	AD_VP	Executive
102	Lex	De Haan	13/01/2001	AD_VP	Executive
103	Alexander	Hunold	03/01/2006	IT_PROG	IT
104	Bruce	Ernst	21/05/2007	IT_PROG	IT
105	David	Austin	25/06/2005	IT_PROG	IT
106	Valli	Pataballa	05/02/2006	IT_PROG	IT
107	Diana	Lorentz	07/02/2007	IT_PROG	IT
108	Nancy	Greenberg	17/08/2002	FI_MGR	Finance
109	Daniel	Faviet	16/08/2002	FI_ACCOUNT	Finance
110	John	Chen	28/09/2005	FI_ACCOUNT	Finance
111	Ismael	Sciarra	30/09/2005	FI_ACCOUNT	Finance
112	Jose Manuel	Urman	07/03/2006	FI_ACCOUNT	Finance

Subconsultas en INSERT

```
--EJEMPLO
INSERT INTO TEST ( SELECT * FROM TEST_SOURCE );

INSERT INTO TEST (CAMPO1_TXT, CAMPO2_NUM)
( SELECT CAMPO1_TXT_SC, CAMPO2_NUM_SC
    FROM TEST_SOURCE
   WHERE CAMPO1_TXT_SC = '777' );

INSERT INTO TEST (CAMPO1_TXT, CAMPO2_NUM, CAMPO3_DAT)
( SELECT CAMPO1_TXT_SC, CAMPO2_NUM_SC, NULL
    FROM TEST_SOURCE );

INSERT INTO TEST
( SELECT CAMPO1_TXT_SC, CAMPO2_NUM_SC, SYSDATE + 5
    FROM TEST_SOURCE );
```

Subconsultas en UPDATE

```
--EJEMPLO
UPDATE TEST
   SET CAMPO1_TXT = ( SELECT CAMPO1_TXT_SC
                        FROM TEST_SOURCE
                       WHERE CAMPO2_NUM_SC = 123 )
 WHERE CAMPO2_NUM = 999;
```