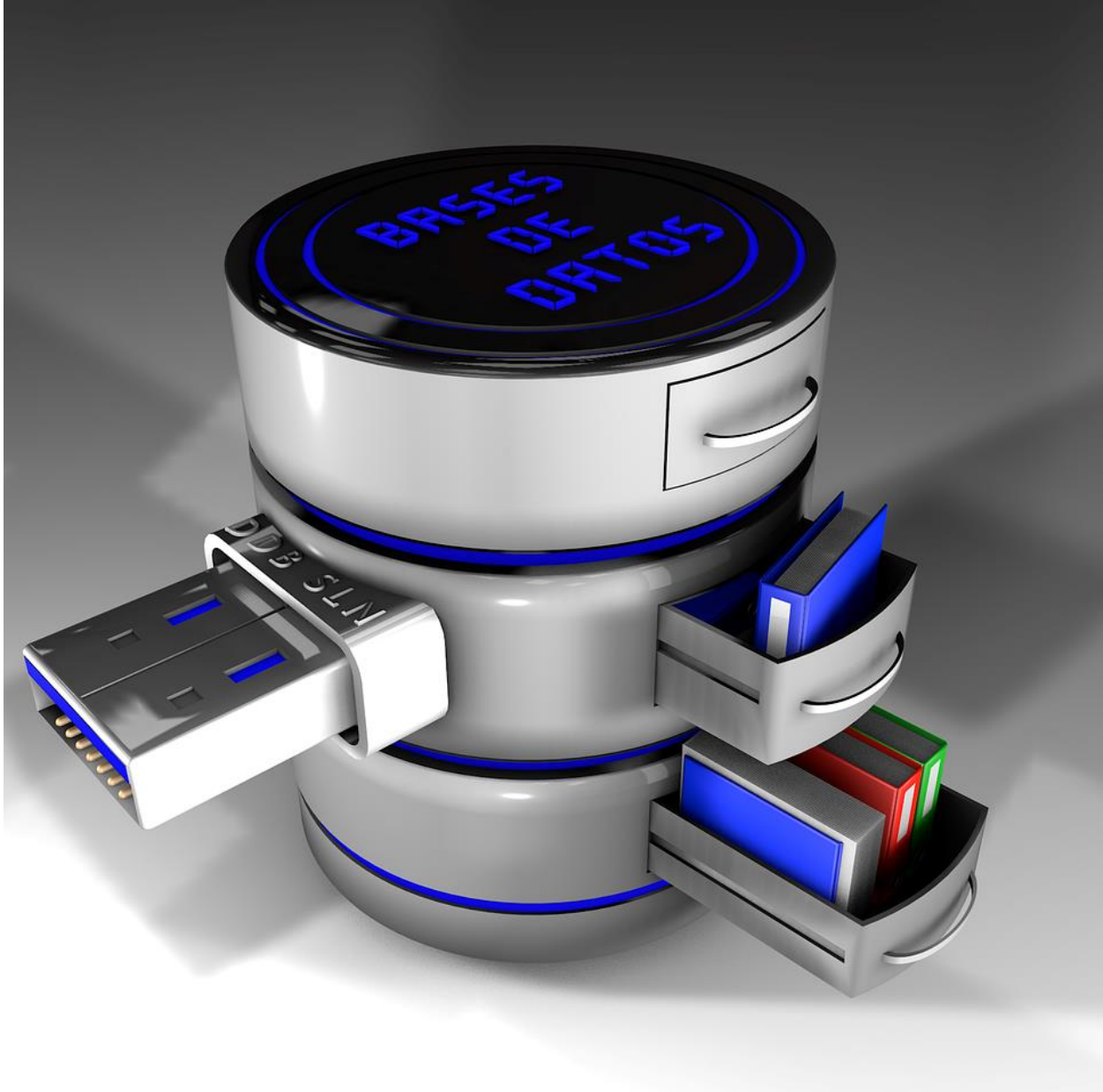
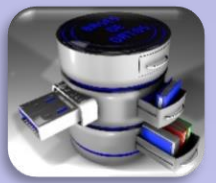
UNITAT DIDÀCTICA 8

Introducció a SQL. Consultes bàsiques

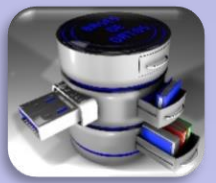


Índex

1. Origen llenguatge SQL
2. Estructura de la base de dades sobre la qual es treballarà.
3. Clàusula SELECT
4. DESCRIBE
5. Clàusula WHERE
6. Conéixer les regles de prioritats



Índex



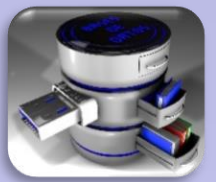
Origen del llenguatge SQL

En 1970, E. F. Codd proposa el model relacional i associat a aquest, un subllenguatge d'accés a les dades basat en el càlcul de predicats.

Basant-se en aquestes idees, els laboratoris d'IBM van definir el llenguatge SEQUEL (Structured English Query Language) que més tard va ser àmpliament implementat pel SGBD experimental System R, desenvolupat en 1977 també per IBM.

No obstant això, va ser Oracle qui ho va introduir per primera vegada en 1979 en un producte comercial.

SEQUEL va acabar sent el predecessor de SQL, que és una versió evolucionada del primer. SQL passa a ser el llenguatge per excel·lència dels diversos sistemes de gestió de bases de dades relacionals

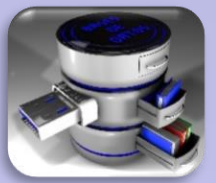


Origen del llenguatge SQL

Va ser per fi estandarditzat en 1986 pel ANSI, donant lloc a la primera versió estàndard d'aquest llenguatge, "SQL-86" o "SQL1". A l'any següent aquest estàndard és també adoptat per ISO.

No obstant això, aquest primer estàndard no cobria totes les necessitats dels desenvolupadors i incloïa funcionalitats de definició d'emmagatzematge que es va considerar suprimir-les. Així que, en 1992, es va llançar un nou estàndard ampliat i revisat de SQL anomenat "SQL-92" o "SQL2".

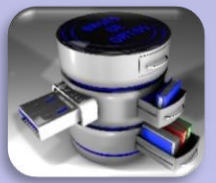
Año	Nombre
1986	SQL-86
1989	SQL-89
1992	SQL-92
1999	SQL:1999
2003	SQL:2003
2006	SQL:2006
2008	SQL:2008
2011	SQL:2011
2016	SQL:2016



Origen del llenguatge SQL

És utilitzat pels SGBD per a realitzar operacions sobre les dades d'una base de dades o sobre l'estructura d'aquesta. Està format per comandos, operadors, clàusules i funcions combinats en sentències per a crear, actualitzar i manipular BD.

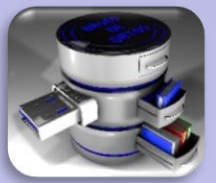
Es denomina estructurat perquè treballa amb conjunts de resultats abstractes com a unitats completes. Un conjunt de resultats és l'esquema bàsic d'una taula: N files x N columnes.



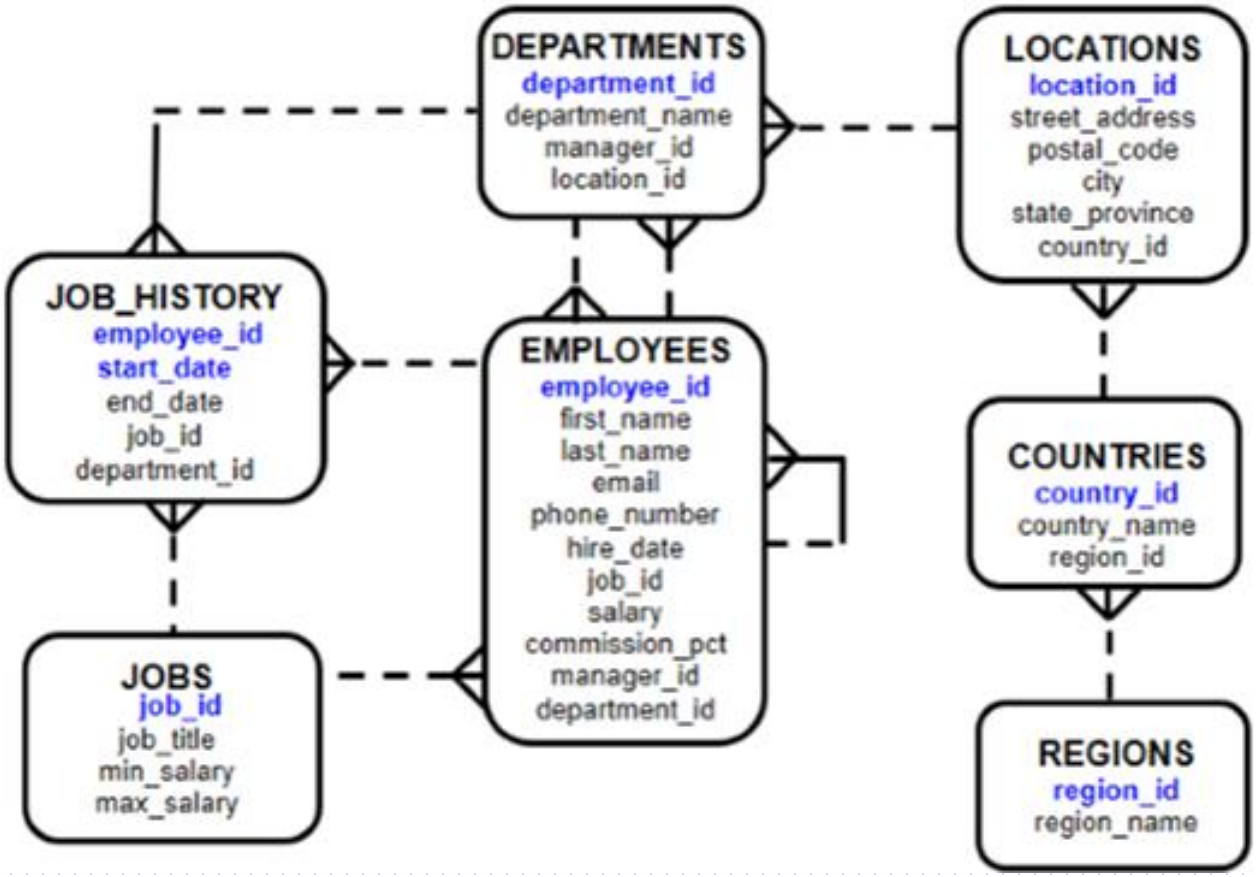
Origen del llenguatge SQL

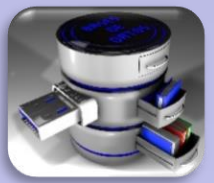
Existeixen tres subllenguatges dins de SQL:

- ✓ Llenguatge de manipulació de dades (DML): Permet a l'usuari consultar dades emmagatzemades en la base de dades, així com actualitzar-la afegint, suprimint o modificant dades.
- ✓ Llenguatge de definició de dades (DDL): Utilitzat per a definir, modificar i esborrar les taules en les quals s'emmagatzemen les dades i les relacions entre aquestes.
- ✓ Llenguatge de control de dades (DCL): Serveix per a definir la protecció i la seguretat de les dades, i la manera de compartir-los concurrentment en un entorn multiusuari.



Estructura de la BD sobre la qual es treballarà

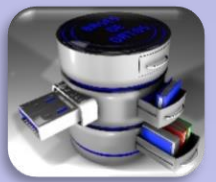




Estructura de la BD sobre la qual es treballarà

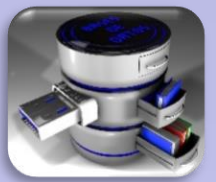
Descripció de les taules:

- ✓ REGIONS conté files que representen una regió, com Amèrica, Àsia, etc.
- ✓ COUNTRIES conté files per a països, que estan associats a una regió.
- ✓ LOCATIONS conté la direcció concreta d'una oficina, magatzem o fàbrica d'una companyia en un país determinat.
- ✓ DEPARTMENTS mostra detalls dels departaments en els quals treballen els empleats. Cada departament pot tindre una relació que represente al gestor del departament en la taula.



Estructura de la BD sobre la qual es treballarà

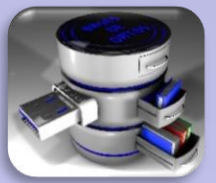
- ✓ EMPLOYEES conté detalls sobre cada empleat que treballa en un departament. Pot ser que alguns empleats no estiguen assignats a cap departament.
- ✓ JOBS conté els tipus de càrrecs que pot tindre cada empleat.
- ✓ JOB_HISTORY conté l'historial del treball dels empleats. Si un empleat canvia de departament dins d'un mateix càrrec o canvia de càrrec dins d'un mateix departament, s'inserirà una nova fila en aquesta taula amb la informació de l'antic càrrec de l'empleat.



Clàusula SELECT

Una sentència **SELECT** recupera informació de la base de dades. Amb una sentència SELECT, es pot fer el següent:

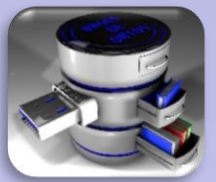
- **Projecció**: seleccione les columnes d'una taula retornades per una consulta. Seleccione tantes columnes com siga necessari.
- **Selecció**: seleccione les files d'una taula retornades per una consulta. Es poden utilitzar diferents criteris per a restringir les files recuperades.
- **Unions**: reunisca les dades emmagatzemades en diferents taules especificant l'enllaç entre elles.



Clàusula SELECT

```
SELECT *|{[DISTINCT] column|expression [alias],...}
FROM    table;
```

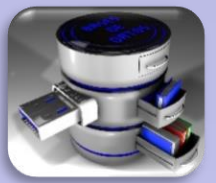
- SELECT: especifica les columnes que es mostraran.
- FROM identifica la taula que conté les columnes que es mostren en la clàusula SELECT.



Clàusula SELECT

```
SELECT *|{[DISTINCT] column|expression [alias],...}
FROM    table;
```

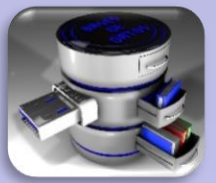
- SELECT
  - * : selecciona totes les columnes.
  - DISTINCT: suprimeix els duplicats.
  - Column|expresión: selecciona la columna o expressió especificada.
  - Alias: proporciona diferents capçaleres de les columnes seleccionades.
- FROM table: especifica la taula que conté les columnes.



Clàusula SELECT

Sintaxis

- | la barra separa opcions de les quals caldrà triar una
- {} el contingut de les claus pot repetir-se
- []el que apareix a l'interior dels corchetes és opcional, pot utilitzar-se o no
- SQL no és *case sensitive*, però les paraules reservades s'escriuran en majúscules.

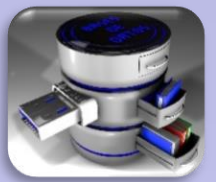


Clàusula SELECT

```
SELECT *
FROM departments;
```

(*) Mostra totes les columnes

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	114	1700
40	Human Resou...	203	2400
50	Shipping	121	1500
60	IT	103	1400

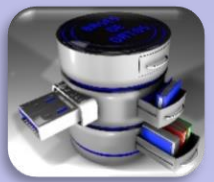


Clàusula SELECT

```
SELECT department_id, department_name
FROM departments;
```

Mostra només les columnas que s'indiquen al SELECT

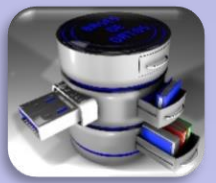
DEPARTMENT_ID	DEPARTMENT_NAME
10	Administration
20	Marketing
30	Purchasing
40	Human Resources
50	Shipping



Clàusula SELECT

Escriptura SQL

- Les sentències SQL es poden introduir en una o més línies.
- Les paraules clau no es poden dividir entre línies o abreujar.
- Les clàusules se solen col·locar en línies independents perquè resulte més fàcil la seua lectura o edició.
- El sagnat s'ha d'utilitzar perquè siga més fàcil de llegir el codi.
- La sentència finalitza amb punt i coma “;”.



Clàusula SELECT

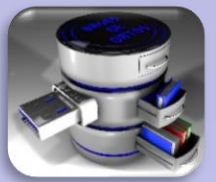
Expressions aritmètiques

Pot ser que necessite modificar la forma en la qual es mostren les dades, realitzar càlculs o consultar casos de possibilitats. Tot això és possible mitjançant les expressions aritmètiques.

Una **expressió aritmètica** pot contindre noms de columna, valors numèrics constants i operadors aritmètics.

Pot utilitzar **operadors aritmètics (+, - , *, /)** en qualsevol clàusula d'una sentència SQL (excepte en la clàusula *FROM).

Nota: amb els tipus de dada DATE i TIMESTAMP, només pot utilitzar els operadors de suma i resta.

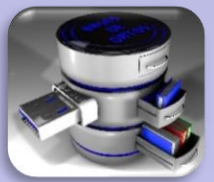


Clàusula SELECT

Expressions aritmètiques

```
SELECT last_name, salary, salary + 300
FROM employees;
```

LAST_NAME	SALARY	SALARY+300
King	24000	24300
Kochhar	17000	17300
De Haan	17000	17300
Hunold	9000	9300
Ernst	6000	6300



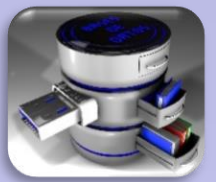
Clàusula SELECT

Expressions aritmètiques

Prioritat d'Operadors

Si una expressió aritmètica conté més d'un operador, la multiplicació i divisió s'avaluen primer. Si els operadors en una expressió tenen la mateixa prioritat, l'avaluació es realitza d'esquerra a dreta.

Pot utilitzar els parèntesis per a forçar l'expressió que s'inclou entre parèntesi perquè s'avalue primer.



Clàusula SELECT

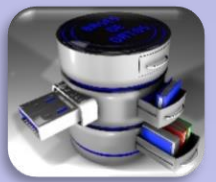
Expressions aritmètiques

Regles de Prioritat dels operadors

La multiplicació i divisió es produeixen abans de la suma i la resta.

Els operadors de la mateixa prioritat s'avaluen d'esquerra a dreta.

Els parèntesis s'utilitzen per a substituir la prioritat per defecte o per a aclarir la sentència.



Clàusula SELECT

Expressions aritmètiques

Prioritat d'Operadors

```
SELECT last_name, salary, 12*salary + 100
FROM employees;
```

(1)

```
SELECT last_name, salary, 12*(salary + 100)
FROM employees;
```

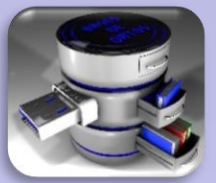
(2)

(1)

LAST_NAME	SALARY	12*SALARY+100
King	24000	288100
Kochhar	17000	204100
De Haan	17000	204100
Hunold	9000	108100
Ernst	6000	72100

(2)

LAST_NAME	SALARY	12*(SALARY+100)
King	24000	289200
Kochhar	17000	205200
De Haan	17000	205200
Hunold	9000	109200
Ernst	6000	73200



Clàusula SELECT

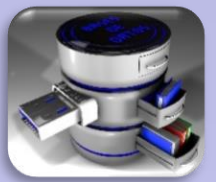
Valor nul

Si una fila no te un valor de dades per a una columna concreta, es diu que aquest valor és **nul** o que conté un valor nul.

Un valor nul és un valor que no està disponible, sense assignar, desconegut o que no és aplicable. Un valor nul no és el mateix que un zero o un espai en blanc. El zero és un número i l'espai en blanc és un caràcter.

Les columnes poden contindre valors nuls a excepció de les que tinguen restriccions NOT NULL i PRIMARY KEY.

Si qualsevol valor de columna en una expressió aritmètica és nul, el resultat és nul.

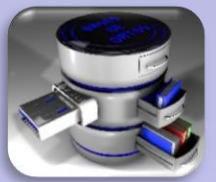


Clàusula SELECT

Valor nul

```
SELECT last_name, salary, job_id, commission_pct
FROM employees;
```

LAST_NAME	SALARY	JOB_ID	COMMISSION_PCT
Matos	2600	ST_CLERK	(null)
Vargas	2500	ST_CLERK	(null)
Russell	14000	SA_MAN	0,4
Partners	13500	SA_MAN	0,3
Errazuriz	12000	SA_MAN	0,3

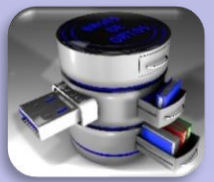


Clàusula SELECT

Valor nul

```
SELECT last_name, salary + (salary*commission_pct)
FROM employees;
```

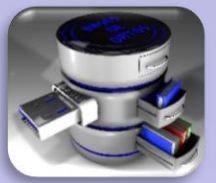
LAST_NAME	SALARY+(SALARY*COMMISSION_PCT)
Davies	(null)
Matos	(null)
Vargas	(null)
Russell	19600
Partners	17550



Clàusula SELECT

Àlies de columna

- canvia el nom d'una capçalera de columna
- És útil per a realitzar càlculs
- Segueix immediatament al nom de columna (també pot ser la paraula clau opcional AS entre el nom de columna i l'àlies)
- Necessita cometes dobles si conté espais o caràcters especials o si és sensible a majúscules/minúscules

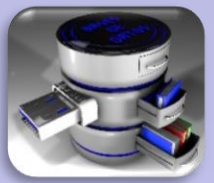


Clàusula SELECT

Àlies de columna

```
SELECT last_name as apellido, first_name "Nombre", department_id departamento
FROM employees;
```

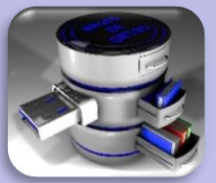
APELLIDO	Nombre	DEPARTAMENTO
King	Steven	90
Kochhar	Neena	90
De Haan	Lex	90
Hunold	Alexander	60
Ernst	Bruce	60



Clàusula SELECT

Operador de concatenació

- enllaça columnes o cadenes de caràcters a altres columnes
- es representa amb dues barres verticals (||)
- crea un columna resultant que és una expressió de carácter
- Si concatena un valor nul amb una cadena de caràcters, el resultat és una cadena de caràcters

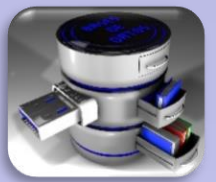


Clàusula SELECT

Operador de concatenació

```
SELECT last_name || first_name "Nombre Completo"
FROM employees;
```

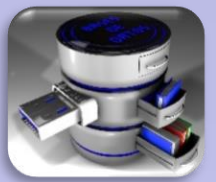
Nombre Completo
AbelEllen
AndeSundar
AtkinsonMozhe
AustinDavid
BaerHermann



Clàusula SELECT

Cadenes de caràcters literals

- Un literal és un caràcter, un número o una data es que inclou en la sentència SELECT
- Els valors literals de caràcters i data s'han d'incloure entre cometes simples
- Cada cadena de caràcters és l'eixida una vegada per a cada fila retornada

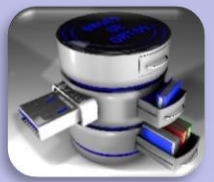


Clàusula SELECT

Cadenes de caràcters literals

```
SELECT last_name ||', ' || first_name "Apellido, Nombre"
FROM employees;
```

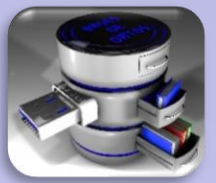
Apellido, Nombre
Abel, Ellen
Ande, Sundar
Atkinson, Mozhe
Austin, David
Baer, Hermann



Clàusula SELECT

Files duplicades

- Llevat que indique el contrari, SQL mostra els resultats d'una consulta sense eliminar les files duplicades.
- Per a eliminar files duplicades en el resultat, incloga la paraula clau DISTINCT en la clàusula SELECT immediatament després de la paraula clau SELECT.

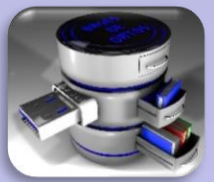


Clàusula SELECT

Files duplicades

```
SELECT DISTINCT department_id
FROM employees;
```

DEPARTMENT_ID
100
30
(null)
90
20



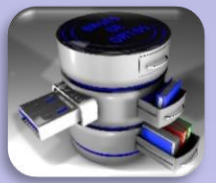
Clàusula SELECT

Files duplicades

- Pot especificar diverses columnes després del cualificador DISTINCT. El cualificador DISTINCT afecta a totes les columnes seleccionades i el resultat és cada combinació diferent de columnes.

```
SELECT DISTINCT department_id, job_id
FROM employees;
```

DEPARTMENT_ID	JOB_ID
110	AC_ACCOUNT
90	AD_VP
50	ST_CLERK
80	SA_REP



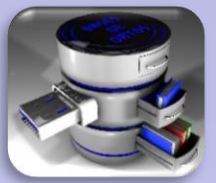
DESCRIBE

```
DESC[RIBE] tablename
```

Pot mostrar l'estructura d'una taula mitjançant el comando DESCRIBE.

El comando mostra els noms de columnes i els tipus de dades i indica si una columna ha de contindre dades (és a dir, si la columna té una restricció NOT NULL).

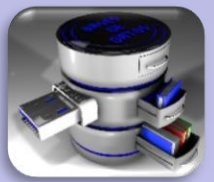
En la sintaxi, *table name* és el nom de qualsevol taula existent, vista o sinònim accessible a l'usuari.



DESCRIBE

```
DESCRIBE departments;
```

```
Nombre           Nulo     Tipo
---------------- -------- ------------
DEPARTMENT_ID    NOT NULL NUMBER(4)
DEPARTMENT_NAME  NOT NULL VARCHAR2(30)
MANAGER_ID                NUMBER(6)
LOCATION_ID               NUMBER(4)
```

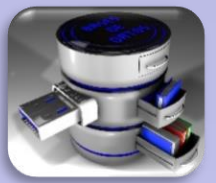



Clàusula WHERE

Pot restringir les files que retorna la consulta en utilitzar la clàusula WHERE.

Una clàusula **WHERE** conté una condició que s'ha de complir. Si la condició és vertadera, es retornarà la fila que complisca amb la condició.

```
SELECT *|{[DISTINCT] column|expression [alias],...}
FROM   table
[WHERE condition(s)];
```

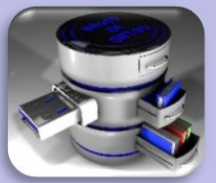



Clàusula WHERE

```
SELECT *|{[DISTINCT] column|expression [alias],...}
FROM   table
[WHERE condition(s)];
```

En la sintaxi:

- **WHERE** restringeix la consulta a files que complisquen amb una condició.
- **condition** està compost per noms de columna, expressions, constants i un operador de comparació. Una condició especifica una combinació d'una o més expressions i operadors lògics (booleans) i retorna un valor de TRUE, FALSE o UNKNOWN.

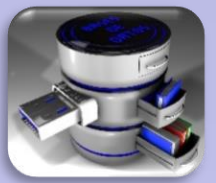


Clàusula WHERE

```
SELECT *|{[DISTINCT] column|expression [alias],...}
FROM   table
[WHERE condition(s)];
```

La clàusula WHERE pot comparar valors en columnes, literals, expressions aritmètiques o funcions. Consta de tres elements:

- Nom de la columna
- Condició de comparació
- Nom de la columna, constant o llista de valors

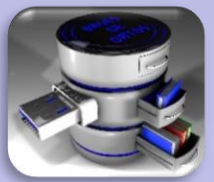


Clàusula WHERE

```
SELECT *|{[DISTINCT] column|expression [alias],...}
FROM   table
[WHERE condition(s)];
```

```
SELECT employee_id, last_name, job_id, department_id
FROM employees
WHERE department_id=90;
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90



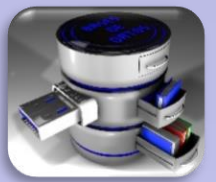
Clàusula WHERE

Les dates i cadenes de caràcters de la clàusula WHERE s'han d'incloure entre cometes simples (' '). No obstant això, les constants numèriques no s'han d'incloure entre cometes simples.

Totes les cerques de caràcters són sensibles a majúscules/minúscules.

```
SELECT employee_id, last_name, job_id, department_id
FROM employees
WHERE last_name LIKE 'Whalen';
```

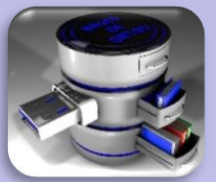
EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
200	Whalen	AD_ASST	10



Clàusula WHERE

```
SELECT employee_id, last_name, job_id, department_id
FROM employees
WHERE hire_date='17/06/03';
```

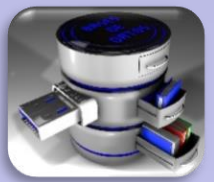
EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90



Clàusula WHERE

Operadors de comparació

Operador	Significat
=	Igual que
>	Major que
>=	Major o igual que
<	Menor que
<=	Menor o igual que
<>	Distint
BETWEEN … AND …	Entre dos valors (inclosos)
IN (valors)	Coincideix amb algún dels valors
LIKE	Coincideix amb un patró
IS NULL	És un valor null



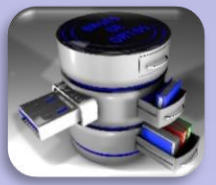
Clàusula WHERE

```
SELECT employee_id, last_name, job_id, department_id
FROM employees
WHERE salary>15000;
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90

```
SELECT employee_id, last_name, job_id, department_id,salary
FROM employees
WHERE salary BETWEEN 9000 AND 10000;
```

EMPLOY...	LAST_NAME	JOB_ID	DEPARTMENT_ID	SALARY
103	Hunold	IT_PROG	60	9000
109	Faviet	FI_ACCOUNT	100	9000
150	Tucker	SA_REP	80	10000
151	Bernstein	SA_REP	80	9500



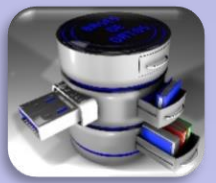
Clàusula WHERE

```
SELECT employee_id, last_name, job_id, department_id,salary
FROM employees
WHERE salary IN(9000,10000);
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID	SALARY
103	Hunold	IT_PROG	60	9000
109	Faviet	FI_ACCOUNT	100	9000
150	Tucker	SA_REP	80	10000
152	Hall	SA_REP	80	9000

```
SELECT employee_id, last_name, job_id, department_id,commission_pct
FROM employees
WHERE commission_pct IS NULL;
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID	COMMISSION_PCT
100	King	AD_PRES	90	(null)
101	Kochhar	AD_VP	90	(null)
102	De Haan	AD_VP	90	(null)



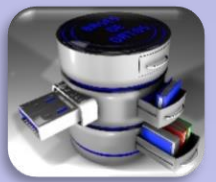
Clàusula WHERE

Coincidència de Patrons mitjançant l'Operador LIKE

Pot ser que no sempre conega el valor exacte que ha de buscar. Pot seleccionar files que coincidisquen amb un patró de caràcters utilitzant la condició LIKE. Es fa referència a l'operació de coincidència de patró de caràcters com a cerca amb comodins. Per a crear la cadena de cerca es poden utilitzar dos símbols:

% →Representa qualsevol seqüència de zero o n caràcters

_ →Representa un únic caràcter



Clàusula WHERE

Coincidència de Patrons mitjançant l'Operador LIKE

```
SELECT first_name, last_name
FROM employees
WHERE last_name LIKE 'S%';
```

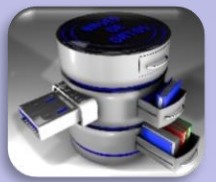
FIRST_NAME	LAST_NAME
Nandita	Sarchand
Ismael	Sciarra
John	Seo
Sarath	Sewall

```
SELECT first_name, last_name
FROM employees
WHERE last_name LIKE '%S';
```

FIRST_NA...	LAST_NA...

```
SELECT first_name, last_name
FROM employees
WHERE last name LIKE '%s';
```

FIRST_NAME	LAST_NAME
Elizabeth	Bates
Karen	Colmenares
Curtis	Davies
Timothy	Gates



Clàusula WHERE

Coincidència de Patrons mitjançant l'Operador LIKE

```
SELECT first_name, last_name
FROM employees
WHERE last_name LIKE '%s%';
```

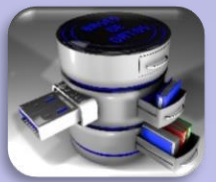
FIRST_NAME	LAST_NAME
Mozhe	Atkinson
David	Austin
Elizabeth	Bates

```
SELECT first_name, last_name
FROM employees
WHERE last_name LIKE '_a%';
```

FIRST_NAME	LAST_NAME
Hermann	Baer
Shelli	Baida
Amit	Banda

```
SELECT first_name, last_name
FROM employees
WHERE last_name LIKE '__a%';
```

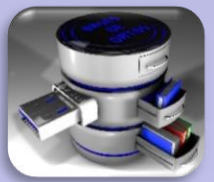
FIRST_NAME	LAST_NAME
Douglas	Grant
Kimberely	Grant
Jennifer	Whalen



Clàusula WHERE

Definició de Condicions mitjançant els Operadors Lògics

Operador	Significat
AND	Retorna TRUE si les condicions que està comprovant són TOTES vertaderes
OR	Retorna TRUE si alguna de les condicions que està comprovant és vertaderes
NOT	Retorna TRUE si la condició es FALSE

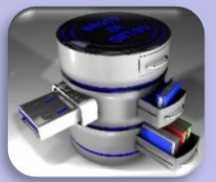


Clàusula WHERE

Definició de Condicions mitjançant els Operadors Lògics

```
SELECT first_name, last_name, job_id, salary
FROM employees
WHERE salary >10000
AND job_id LIKE '%MAN%';
```

FIRST_NAME	LAST_NAME	JOB_ID	SALARY
Den	Raphaely	PU_MAN	11000
John	Russell	SA_MAN	14000
Karen	Partners	SA_MAN	13500
Alberto	Errazuriz	SA_MAN	12000
Gerald	Cambrault	SA_MAN	11000

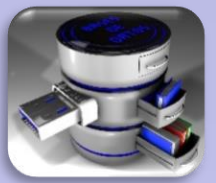


Clàusula WHERE

Definició de Condicions mitjançant els Operadors Lògics

```
SELECT first_name, last_name, job_id, salary
FROM employees
WHERE salary >10000
OR job_id LIKE '%MAN%';
```

FIRST_NAME	LAST_NAME	JOB_ID	SALARY
Steven	King	AD_PRES	24000
Neena	Kochhar	AD_VP	17000
Lex	De Haan	AD_VP	17000
Nancy	Greenberg	FI_MGR	12008
Den	Raphaely	PU_MAN	11000
Matthew	Weiss	ST_MAN	8000

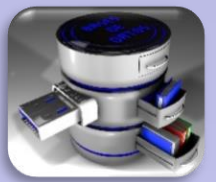


Clàusula WHERE

Definició de Condicions mitjançant els Operadors Lògics

```
SELECT last_name, job_id
FROM EMPLOYEES
WHERE job_id NOT IN ('IT_PROG', 'ST_CLERK');
```

LAST_NAME	JOB_ID
Ande	SA_REP
Baer	PR_REP
Baida	PU_CLERK
Banda	SA_REP
Bates	SA_REP
Bell	SH_CLERK
Bernstein	SA_REP

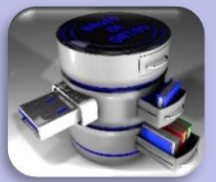


Regles de prioritat

Les regles de prioritat determinen l'ordre en el qual s'avaluen i calculen les expressions. No obstant això, pot substituir l'ordre per defecte utilitzant parèntesi en les expressions que desitge calcular primer.

```
SELECT last_name, job_id, salary
FROM employees
WHERE job_id = 'SA_REP'
OR job_id = 'AD_PRES'
AND salary > 15000;
```

LAST_NAME	JOB_ID	SALARY
King	AD_PRES	24000
Tucker	SA_REP	10000
Bernstein	SA_REP	9500
Hall	SA_REP	9000

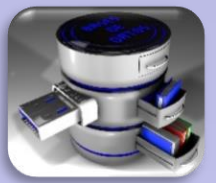


Regles de prioritat

Les regles de prioritat determinen l'ordre en el qual s'avaluen i calculen les expressions. No obstant això, pot substituir l'ordre per defecte utilitzant parèntesi en les expressions que desitge calcular primer.

```
SELECT last_name, job_id, salary
FROM employees
WHERE (job_id = 'SA_REP'
OR job_id = 'AD_PRES')
AND salary > 15000;
```

LAST_NAME	JOB_ID	SALARY
King	AD_PRES	24000

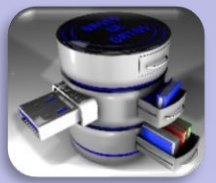


Ordenació de files. ORDER BY

L'ordre de les files retornades en un resultat de consulta no està definit. La clàusula ORDER BY es pot utilitzar per a ordenar les files. No obstant això, si utilitza la clàusula ORDER BY, ha de ser l'última clàusula de la sentència SQL. A més, pot especificar una expressió, un àlies o una posició de columna com la condició d'ordenació.

Sintaxis

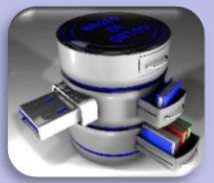
```
SELECT      expr
FROM              Tabla
[WHERE      condition(s)]
[ORDER BY   {column, expr, numeric_position} [ASC|DESC]];
```

Ordenació de files. ORDER BY

```
SELECT last_name, job_id, department_id, hire_date
FROM employees
ORDER BY hire_date;
```

LAST_NAME	JOB_ID	DEPARTMENT_ID	HIRE_DATE
De Haan	AD_VP	90	13/01/01
Gietz	AC_ACCOUNT	110	07/06/02
Baer	PR_REP	70	07/06/02
Mavris	HR_REP	40	07/06/02
Higgins	AC_MGR	110	07/06/02
Faviet	FI_ACCOUNT	100	16/08/02
Greenberg	FI_MGR	100	17/08/02
Raphaely	PU_MAN	30	07/12/02

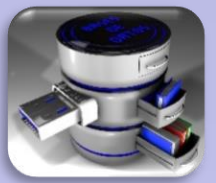


Ordenació de files. ORDER BY

Ordre ascendent

```
SELECT last_name, job_id, department_id, hire_date
FROM employees
ORDER BY hire_date asc;
```

LAST_NAME	JOB_ID	DEPARTMENT_ID	HIRE_DATE
De Haan	AD_VP	90	13/01/01
Gietz	AC_ACCOUNT	110	07/06/02
Baer	PR_REP	70	07/06/02
Mavris	HR_REP	40	07/06/02
Higgins	AC_MGR	110	07/06/02
Faviet	FI_ACCOUNT	100	16/08/02
Greenberg	FI_MGR	100	17/08/02
Raphaely	PU_MAN	30	07/12/02

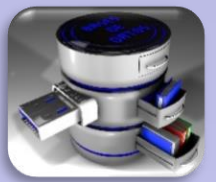


Ordenació de files. ORDER BY

Ordre descendent

```
SELECT last_name, job_id, department_id, hire_date
FROM employees
ORDER BY hire_date desc;
```

LAST_NAME	JOB_ID	DEPARTMENT_ID	HIRE_DATE
Kumar	SA_REP	80	21/04/08
Banda	SA_REP	80	21/04/08
Ande	SA_REP	80	24/03/08
Markle	ST_CLERK	50	08/03/08
Lee	SA_REP	80	23/02/08
Philtanker	ST_CLERK	50	06/02/08

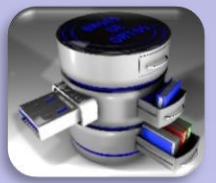


Ordenació de files. ORDER BY

Altres formes d'ordenar

```
SELECT last_name, job_id, department_id departamento, hire_date
FROM employees
ORDER BY departamento desc;
```

LAST_NAME	JOB_ID	DEPARTAMENTO	HIRE_DATE
Faviet	FI_ACCOUNT	100	16/08/02
Greenberg	FI_MGR	100	17/08/02
Urman	FI_ACCOUNT	100	07/03/06
King	AD_PRES	90	17/06/03
Kochhar	AD_VP	90	21/09/05
De Haan	AD_VP	90	13/01/01
Russell	SA_MAN	80	01/10/04
Johnson	SA_REP	80	04/01/08

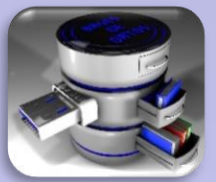


Ordenació de files. ORDER BY

Altres formes d'ordenar

```
SELECT last_name, job_id, department_id departamento, hire_date
FROM employees
ORDER BY 1 desc;
```

LAST_NAME	JOB_ID	DEPARTAMENTO	HIRE_DATE
Zlotkey	SA_MAN	80	29/01/08
Whalen	AD_ASST	10	17/09/03
Weiss	ST_MAN	50	18/07/04
Walsh	SH_CLERK	50	24/04/06
Vollman	ST_MAN	50	10/10/05
Vishney	SA_REP	80	11/11/05
Vargas	ST_CLERK	50	09/07/06
Urman	FI_ACCOUNT	100	07/03/06

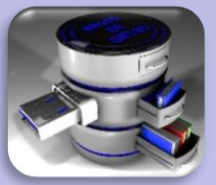


Ordenació de files. ORDER BY

Altres formes d'ordenar

```
SELECT last_name, job_id, department_id departamento, hire_date
FROM employees
ORDER BY 3,1 desc;
```

LAST_NAME	JOB_ID	DEPARTAMENTO	HIRE_DATE
Whalen	AD_ASST	10	17/09/03
Hartstein	MK_MAN	20	17/02/04
Fay	MK_REP	20	17/08/05
Tobias	PU_CLERK	30	24/07/05
Raphaely	PU_MAN	30	07/12/02
Khoo	PU_CLERK	30	18/05/03
Himuro	PU_CLERK	30	15/11/06
Colmenares	PU_CLERK	30	10/08/07
Baida	PU_CLERK	30	24/12/05



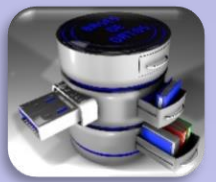
Variable de substitució

Fins ara les sentències SQL s'han executat amb columnes i condicions predeterminades i els seus valors. Suposem que desitja realitzar una consulta que mostre els empleats amb diferents càrrecs

Si s'utilitza una variable de substitució en lloc dels valors exactes en la clàusula WHERE, pot executar la mateixa consulta per a diferents valors.

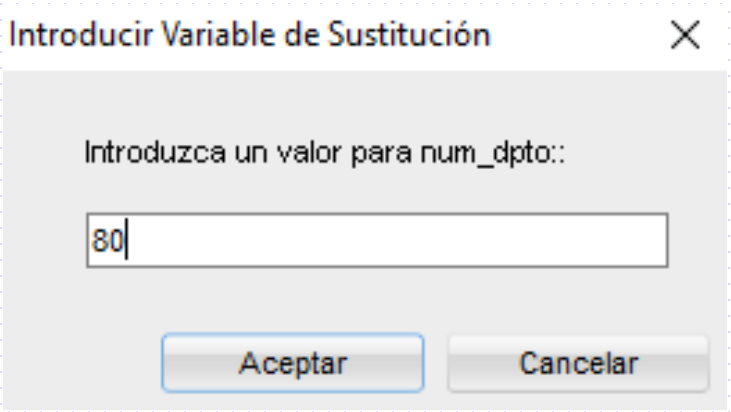
Pot utilitzar variables de substitució d'un sol ampersand (&) per a emmagatzemar valors temporalment.

També pot predefinir variables mitjançant el comando DEFINE. DEFINE crea i assigna un valor a una variable.

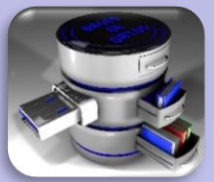


Variable de substitució

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

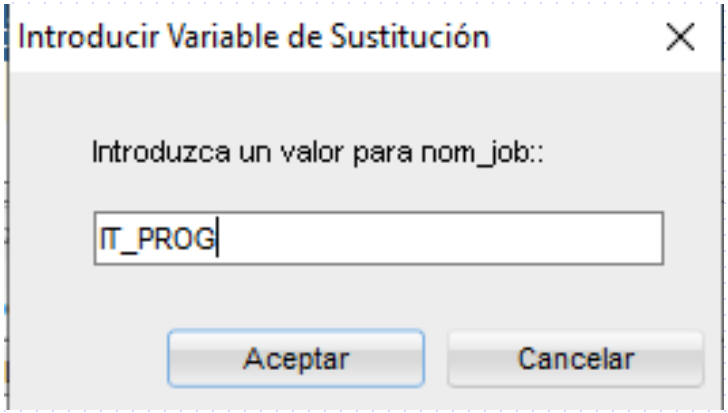


EMPLOY...	LAST_NAME	SALARY	DEPARTMENT_ID
145	Russell	14000	80
146	Partners	13500	80
147	Errazuriz	12000	80
148	Cambrault	11000	80
149	Zlotkey	10500	80
150	Tucker	10000	80
151	Bernstein	9500	80

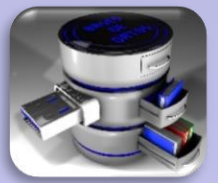


Variable de substitució

```
SELECT employee_id, last_name, job_id, department_id
FROM employees
WHERE job_id = '&nom_job';
```



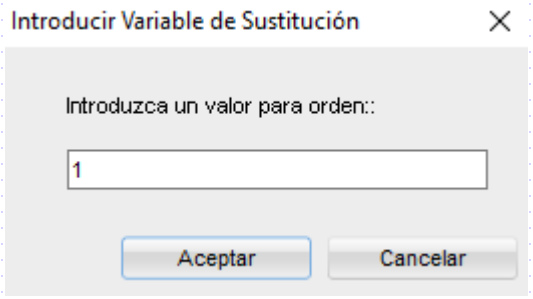
EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
103	Hunold	IT_PROG	60
104	Ernst	IT_PROG	60
105	Austin	IT_PROG	60
106	Pataballa	IT_PROG	60
107	Lorentz	IT_PROG	60



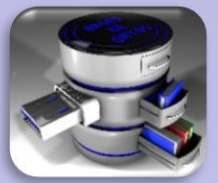
Variable de substitució

No sols pot utilitzar les variables de substitució en la clàusula *WHERE d'una sentència SQL, sinó també com a substitució de noms de columna, expressions o text

```
SELECT employee_id, last_name, job_id, department_id
FROM employees
WHERE job_id = 'IT_PROG'
ORDER BY &orden;
```



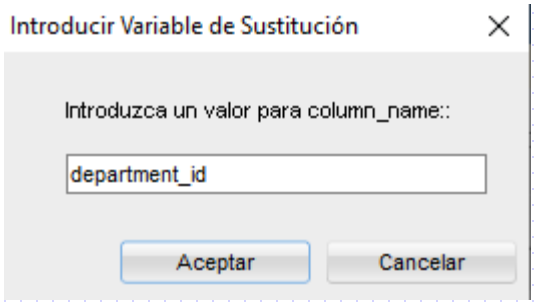
EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
103	Hunold	IT_PROG	60
104	Ernst	IT_PROG	60
105	Austin	IT_PROG	60
106	Pataballa	IT_PROG	60
107	Lorentz	IT_PROG	60



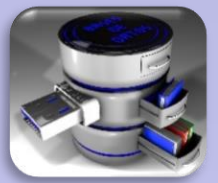
Variable de substitució

Pot utilitzar la variable de substitució de dues ampersands (&&) si desitja reutilitzar el valor de la variable sense preguntar sempre a l'usuari. L'usuari visualitza la sol·licitud del valor només una vegada

```
SELECT employee_id, last_name, job_id, &&column_name
FROM employees
ORDER BY &column_name;
```



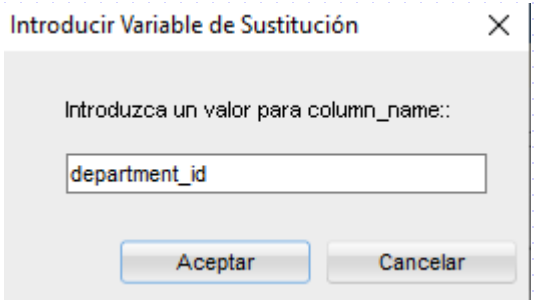
EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
200	Whalen	AD_ASST	10
201	Hartstein	MK_MAN	20
202	Fay	MK_REP	20
114	Raphaely	PU_MAN	30
115	Khoo	PU_CLERK	30
116	Baida	PU_CLERK	30



Variable de substitució

Pot utilitzar la variable de substitució de dues ampersands (&&) si desitja reutilitzar el valor de la variable sense preguntar sempre a l'usuari. L'usuari visualitza la sol·licitud del valor només una vegada

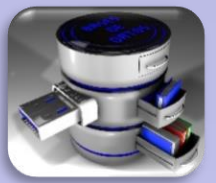
```
SELECT employee_id, last_name, job_id, &&column_name
FROM employees
ORDER BY &column_name;
```



EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
200	Whalen	AD_ASST	10
201	Hartstein	MK_MAN	20
202	Fay	MK_REP	20
114	Raphaely	PU_MAN	30
115	Khoo	PU_CLERK	30
116	Baida	PU_CLERK	30

Per a netejar el valor de la variable, utilitzem el comando UNDEFINE

UNDEFINE nom_variable;

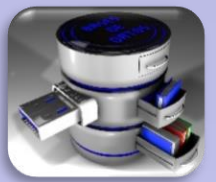


Funcions

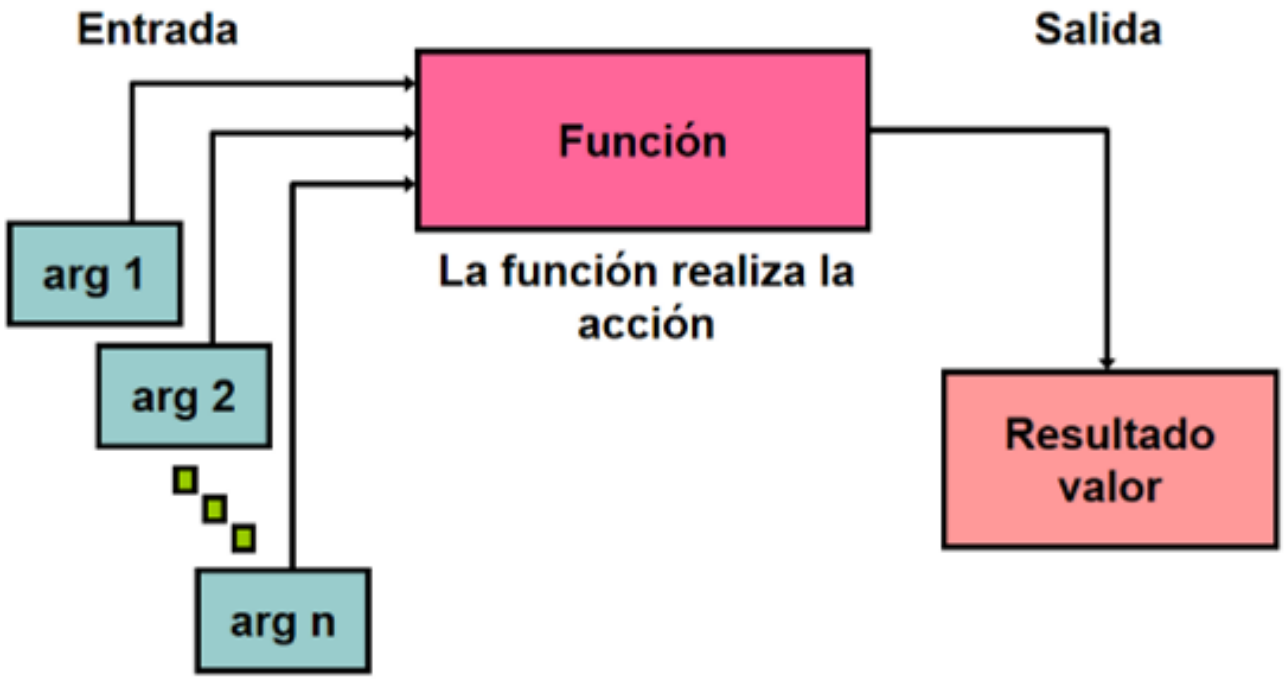
Les funcions són una característica molt potent de SQL. Es pot utilitzar per a realitzar les següents accions:

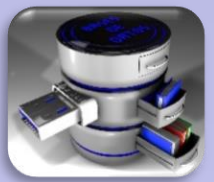
- Realitzar càlculs en les dades
- Modificar elements de dades individuals
- Manipular l'eixida per a grups de files
- Formatar dates i números per a la seua visualització
- Convertir tipus de dada de columna

Algunes vegades, les funcions SQL prenen arguments i sempre retornen un valor.



Funcions

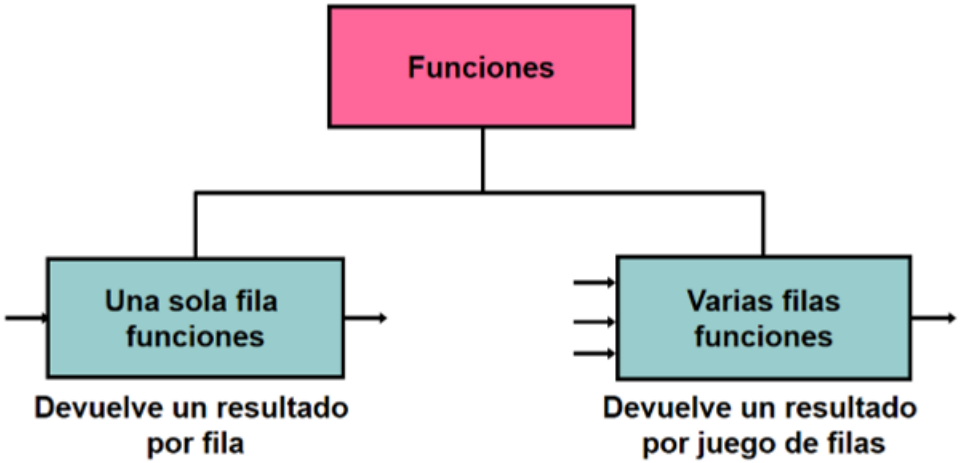

Entrada
Salida
Función
arg 1
arg 2
arg n
La función realiza la acción
Resultado valor

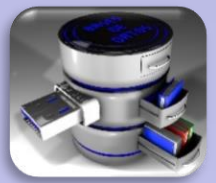


Funcions

Hi ha dos tipus de grups de funcions:

- Funcions d'una sola fila
- Funcions de diverses files



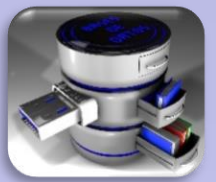


Funcions

Funcions d'una Sola Fila

Aquestes funcions funcionen només en files úniques i retornen un resultat per fila. Existeixen diferents tipus de funcions d'una sola fila. En aquesta lliçó s'aborden els següents temes:

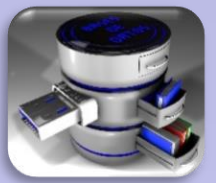
- Caràcter
- Número
- Data
- Conversió
- General



Funcions

Funcions de Diverses Files

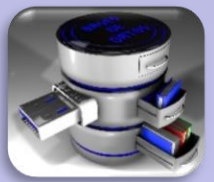
Les funcions poden manipular grups de files per a proporcionar un resultat per grup de files. Aquestes funcions també es coneixen com a funcions de grup (es veuran més endavant)



Funcions d'una sola fila

Les funcions d'una sola fila s'utilitzen per a manipular elements de dades. Accepten un o diversos arguments i retornen un valor per a cada fila retornada per la consulta. Un argument pot ser un dels següents elements:

- Constant proporcionada per l'usuari
- Valor de variable
- Nom de columna
- Expressions



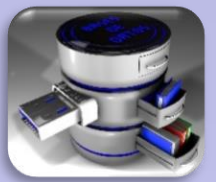
Funcions d'una sola fila

Les característiques de les funcions d'una sola fila són:

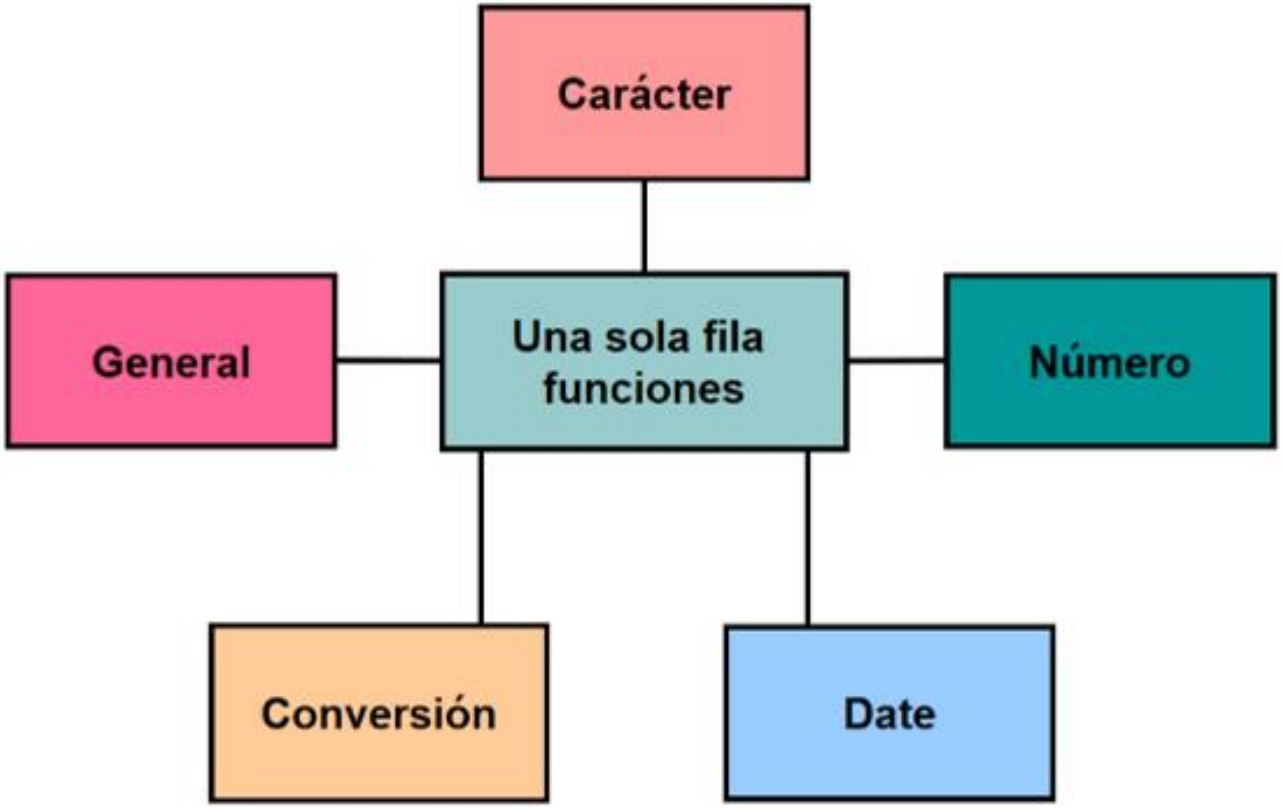
- Actuar en cada fila retornada en la consulta
- Retornar un resultat per fila
- Possibilitat de retornar un valor de dades d'un tipus diferent al que es fa referencia
- Possibilitat d'esperar un o més arguments
- Es poden utilitzar en clàusules SELECT, WHERE i ORDER BY; possibilitat de anidament.

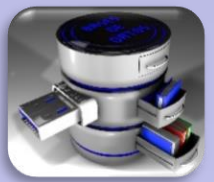
Sintaxis:

```
function_name [(arg1, arg2,...)]
```

Funcions d'una sola fila

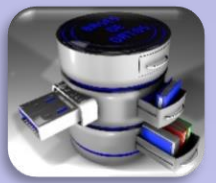




Funcions d'una sola fila

Tractarem les següents funcions d'una sola fila:

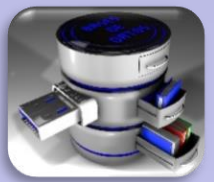
- ✓ **Funcions de caràcter**: accepten l'entrada de caràcters i poden retornar valors de número i de caràcter.
- ✓ **Funcions numèriques**: accepten valors d'entrada i retornen valors numèrics.
- ✓ **Funcions de data**: operen en valors del tipus de dada DATE. (Totes les funcions de data retornen un valor de tipus de dada DATE excepte la funció MONTHS_BETWEEN, que retorna un número.)



Funcions d'una sola fila

Tractarem les següents funcions d'una sola fila:

- ✓ **Funcions de caràcter**: accepten l'entrada de caràcters i poden retornar valors de número i de caràcter.
- ✓ **Funcions de conversió:** Converteixen un valor d'un tipus de dada a un altre
- ✓ **Funcions generals:**
  - NVL
  - NVL2
  - NULLIF
  - COALESCE
  - CASE
  - DECODE

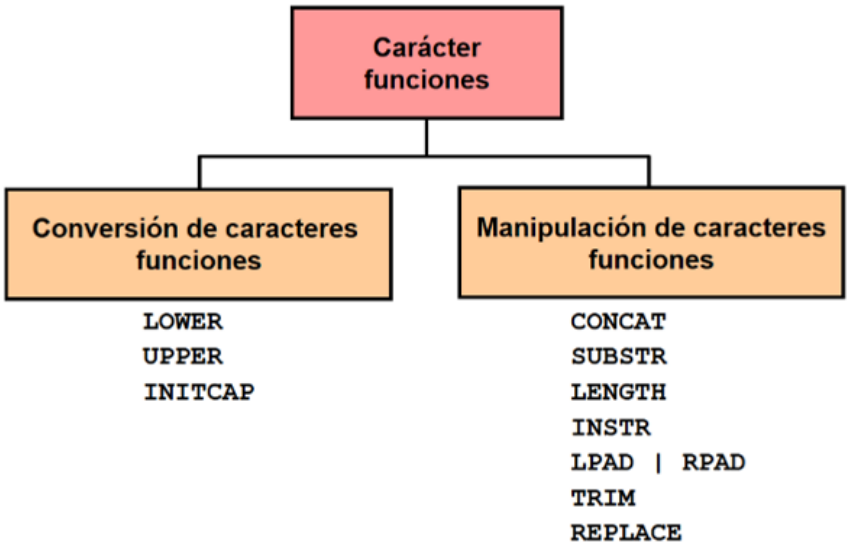


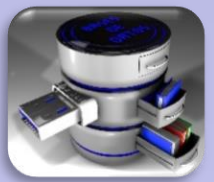
Funcions d'una sola fila

FUNCIONS DE CARÀCTER

Les funcions de caràcter d'una sola fila accepten les dades de caràcters com a entrada i poden retornar valors numèrics i de caràcter. Les funcions de caràcter es poden dividir en:

- Funcions de conversió de caràcters
- Funcions de manipulació de caràcters





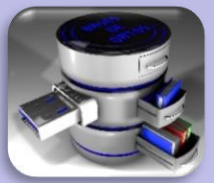
Funcions d'una sola fila

FUNCIONS DE CARÀCTER. Funcions de conversió de caràcters

LOWER, UPPER i INITCAP són les tres funcions de conversió de caràcters.

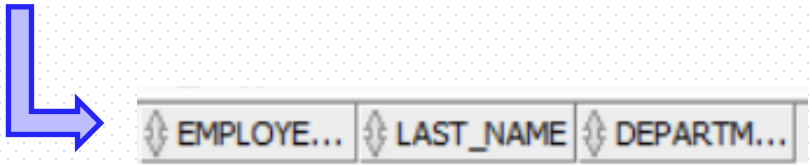
- LOWER: converteix les cadenes de caràcters en majúscules o en majúscules/minúscules a minúscules.
- UPPER: converteix les cadenes de caràcters en minúscula o en majúscules/minúscules a majúscules.
- INITCAP: converteix la primera lletra de cada paraula a majúscules i la resta de les lletres a minúscules.

Función	Resultado
LOWER('SQL Course')	sql course
UPPER('SQL Course')	SQL COURSE
INITCAP('SQL Course')	Sql Course



Funcions d'una sola fila

```
SELECT employee_id, last_name, department_id
FROM employees
WHERE last_name LIKE 'higgins';
```

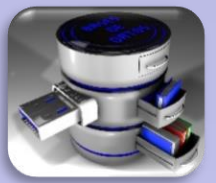


EMPLOYE...	LAST_NAME	DEPARTM...

```
SELECT employee_id, last_name, department_id
FROM employees
WHERE LOWER(last_name) LIKE 'higgins';
```

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
205	Higgins	110

converteix l'atribut last_name a minúscula i el compara amb el text



Funcions d'una sola fila

FUNCIONS DE CARÀCTER. Funcions de conversió de caràcters

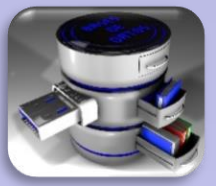
```
SELECT employee_id, last_name, department_id
FROM employees
WHERE last_name LIKE 'higgins';
```

EMPLOYE...	LAST_NAME	DEPARTM...

```
SELECT employee_id, last_name, department_id
FROM employees
WHERE LOWER(last_name) LIKE 'higgins';
```

converteix l'atribut last_name a minúscula i el compara amb el text

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
205	Higgins	110



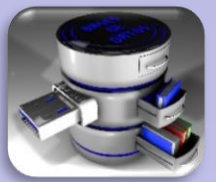
Funcions d'una sola fila

FUNCIONS DE CARÀCTER. Funcions de conversió de caràcters

L'exemple de la diapositiva mostra el número d'empleat, nom i número de departament de l'empleat Higgins:

La clàusula WHERE de la primera sentència SQL especifica el nom de l'empleat com higgins. Pel fet que totes les dades de la taula EMPLOYEES estan emmagatzemats correctament, el nom higgins no troba cap coincidència en la taula i no se selecciona cap fila.

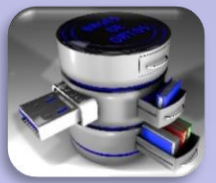
La clàusula WHERE de la segona sentència SQL especifica que el nom de l'empleat de la taula EMPLOYEES es compara amb higgins, convertint la columna LAST_NAME a minúscules per a poder comparar-la. Ja que tots dos noms no estan en minúscules, s'ha trobat una coincidència i s'ha seleccionat una fila.



Funcions d'una sola fila

FUNCIONS DE CARÀCTER. Funcions de manipulació de caràcters

Función	Resultado
CONCAT('Hello', 'World')	HelloWorld
SUBSTR('HelloWorld',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

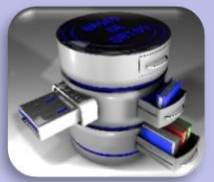


Funcions d'una sola fila

FUNCIONS DE CARÀCTER. Funcions de manipulació de caràcters

CONCAT, SUBSTR, LENGTH, INSTR, LPAD, RPAD y TRIM son funcions de manipulació de caràcters

- CONCAT: uneix els valors. (Només es poden utilitzar dos paràmetres amb CONCAT).
- SUBSTR: extrau una cadena d'una longitud determinada
- LENGTH: mostra la longitud d'una cadena com un valor numèric
- INSTR: obté la posició numèrica d'un caràcter determinat
- LPAD: retorna una expressió amb farciment a l'esquerra de caràcters n amb una expressió de caràcters
- RPAD: retorna una expressió amb farciment a la dreta de caràcters n amb una expressió de caràcters
- TRIM: retalla els caràcters finals o d'encapçalat (o tots dos) d'una cadena de caràcters (si trim_character o trim_source és un caràcter literal, ha d'incloure'l entre cometes simples)

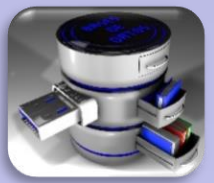


Funcions d'una sola fila

FUNCIONS DE CARÀCTER. Funcions de manipulació de caràcters

```
SELECT employee_id, CONCAT(first_name, last_name) name, job_id,
       LENGTH (last_name), INSTR(last_name, 'a') "Contiene 'a'?"
FROM employees
WHERE SUBSTR(job_id, 4)= 'REP';
```

EMPLOYEE_ID	NAME	JOB_ID	LENGTH(LAST_NAME)	Contiene 'a'?
150	PeterTucker	SA_REP	6	0
151	DavidBernstein	SA_REP	9	0
152	PeterHall	SA_REP	4	2
153	ChristopherOlsen	SA_REP	5	0
154	NanetteCambrault	SA_REP	9	2
155	OliverTuvault	SA_REP	7	4
156	JanetteKing	SA_REP	4	0



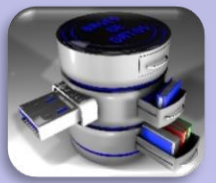
Funcions d'una sola fila

FUNCIONS NUMÈRIQUES

Les funcions numèriques accepten entrades numèriques i retornen valors numèrics. Algunes de les funcions numèriques son:

- ROUND: arredoneix la columna, expressió o valor a n decimals
- TRUNC: trunca la columna, expressió o valor a n decimals
- MOD: Obté la resta d'una divisió

Función	Resultado
ROUND(45.926, 2)	45.93
TRUNC(45.926, 2)	45.92
MOD(1600, 300)	100



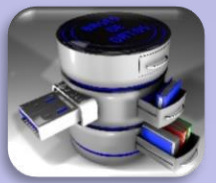
Funcions d'una sola fila

FUNCIONS NUMÈRIQUES

Les funcions numèriques accepten entrades numèriques i retornen valors numèrics. Algunes de les funcions numèriques son:

- ROUND: arredoneix la columna, expressió o valor a n decimals
- TRUNC: trunca la columna, expressió o valor a n decimals
- MOD: Obté la resta d'una divisió

Función	Resultado
ROUND(45.926, 2)	45.93
TRUNC(45.926, 2)	45.92
MOD(1600, 300)	100



Funcions d'una sola fila

FUNCIONS NUMÈRIQUES. ROUND

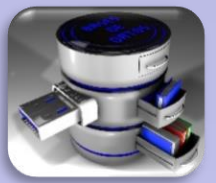
La funció ROUND arredoneix la columna, expressió o valor a n decimals.

ROUND (columna|expresión, n);

Si falta el segon argument o té un valor de 0, el valor s'arredoneix a zero decimals. Si el segon argument té un valor de 2, el valor s'arredoneix a dos decimals. Per contra, si el segon argument és –2, el valor s'arredoneix a dos decimals a l'Esquerra, és a dir, a la centena (arredonits a la unitat més pròxima a 100).

```
SELECT ROUND (45.925,2), ROUND (45.925,0), ROUND (45.925,-1)
FROM DUAL;
```

ROUND(45.925,2)	ROUND(45.925,0)	ROUND(45.925,-1)
45,93	46	50



Funcions d'una sola fila

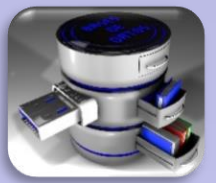
FUNCIONS NUMÈRIQUES. TRUNC

La funció TRUNC funciona amb arguments similars als de la funció ROUND. Si falta el segon argument o té un valor de 0, el valor es trunca a zero decimals. Si el segon argument té un valor de 2, el valor es trunca a dos decimals. Per contra, si el segon argument té un valor de –2, el valor es trunca a dos decimals a l'Esquerra, és a dir, a la centena.

TRUNC (columna|expresión, n);

```
SELECT TRUNC (45.925,2), TRUNC (45.925,0), TRUNC (45.925,-1)
FROM DUAL;
```

TRUNC(45.925,2)	TRUNC(45.925,0)	TRUNC(45.925,-1)
45,92	45	40



Funcions d'una sola fila

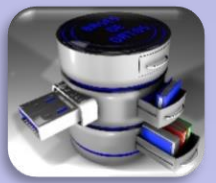
FUNCIONS NUMÈRIQUES. MOD

La funció MOD obté la resta del primer argument (m) dividit entre el segon argument (n). Se sol utilitzar per a determinar si un valor és parell o imparell.

MOD (m, n);

```
SELECT last_name, salary, MOD(salary,5000)
FROM employees;
```

LAST_NAME	SALARY	MOD(SALARY,5000)
King	24000	4000
Kochhar	17000	2000
De Haan	17000	2000



Funcions d'una sola fila

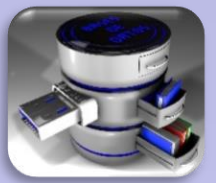
FUNCIÓ SYSDATE

Ús de la Funció SYSDATE

SYSDATE és una funció de data que retorna la data i hora actuals del servidor de base de dades. Pot utilitzar SYSDATE com si utilitzara qualsevol altre nom de columna. Per exemple, pot mostrar la data actual seleccionant SYSDATE d'una taula, per exemple una taula fictícia denominada DUAL.

```
SELECT sysdate
FROM dual;
```

SYSDATE
12/01/22



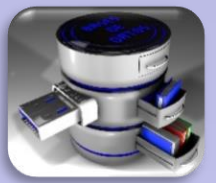
Operadors Aritmètics amb Dates

Pel fet que la base de dades emmagatzema dates com a números, pot realitzar càlculs utilitzant operadors aritmètics com la suma i la resta. Pot agregar i restar constants numèriques i dates.

L'exemple mostra el cognom i el nombre de setmanes durant les quals han treballat tots els empleats del departament 90. Resta la data de contractació de l'empleat de la data actual (SYSDATE) i divideix el resultat entre 7 per a calcular el nombre de setmanes durant les quals ha treballat l'empleat.

```
SELECT last_name, (SYSDATE - hire_date)/7 AS weeks
FROM employees
WHERE department_id=90;
```

LAST_NAME	WEEKS
King	969,202723214285714285714285714285714286
Kochhar	851,059866071428571428571428571428571429
De Haan	1095,631294642857142857142857142857142857



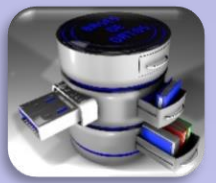
Operadors Aritmètics amb Dates

Podem unir les funcions per a fer una millor lectura de les dades

```
SELECT last_name, ROUND((SYSDATE - hire_date)/7,0) AS weeks
FROM employees
WHERE department_id=90;
```

LAST_NAME	WEEKS
King	969
Kochhar	851
De Haan	1096

setmanes arredonides a la part sencera

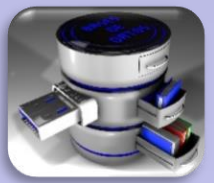


Funcions de Manipulació de Data

Funcions de Manipulació de Data

Les funcions de data funcionen en dates d'Oracle. Totes les funcions de data retornen un valor de tipus data (DATE) excepte MONTHS_BETWEEN, que retorna un valor numèric.

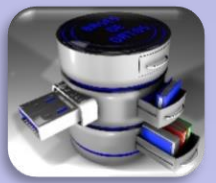
- **MONTHS_BETWEEN**(date1, date2): obté el nombre de mesos entre date1 i date2. El resultat pot ser positiu o negatiu. Si date1 és posterior a date2, el resultat és positiu; en cas contrari, el resultat és negatiu. La part del resultat que no siga un enter representa una part del mes.
- **ADD_MONTHS**(date, n): agrega el número n dels mesos de calendari a date. El valor de n ha de ser un sencer i pot ser negatiu.
- **NEXT_DAY**(date, 'char'): obté la data del següent dia de la setmana especificat ('char') que li segueix a date. El valor de char pot ser un número que representa un dia o una cadena de caràcters.
- **LAST_DAY**(date): obté la data de l'últim dia del mes que conté date.



Funcions de Manipulació de Data

Funcions de Manipulació de Data

Función	Resultado
MONTHS_BETWEEN ('01-SEP-95','11-JAN-94')	19.6774194
ADD_MONTHS ('31-JAN-96',1)	'29-FEB-96'
NEXT_DAY ('01-SEP-95','FRIDAY')	'08-SEP-95'
LAST_DAY ('01-FEB-95')	'28-FEB-95'



Funcion TO_CHAR

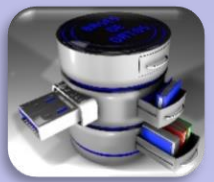
TO_CHAR converteix un tipus de dada de data i hora a un valor e tipus de dada VARCHAR2 amb el format especificat per *format_model*. Un model de format és un caràcter literal que descriu el format de data i hora emmagatzemat en una cadena de caràcters.

Per exemple el format de data i hora '11 – nov – 2020' és 'DD – mon – YYYY'

```
TO_CHAR(date, 'format_model')
```

Instruccions

- ✓ El model de format deu estar entre cometes simples i és sensible a majúscules/minúscules.
- ✓ El model de format pot incloure qualsevol element de format de data vàlid. Cal separar el valor de data del model de format amb una coma.
- ✓ Els noms dels dies i mesos en l'eixida s'emplenen automàticament amb espais en blanc
- ✓ Per a eliminar els espais en blanc o suprimir els zeros inicials, cal utilitzar l'element *fm*



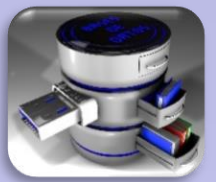
Funcion TO_CHAR

```
SELECT employee_id, TO_CHAR(hire_date, 'MM/YY') "MES AÑO"
FROM employees
WHERE last_name LIKE 'Higgins';
```

EMPLOYEE_ID	MES AÑO
205	06/02

```
SELECT employee_id, TO_CHAR(hire_date, 'DD-MONTH') "MES AÑO"
FROM employees
WHERE last_name LIKE 'Higgins';
```

EMPLOYEE_ID	MES AÑO
205	07-JUNIO



Funcion TO_CHAR

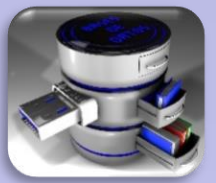
```
SELECT employee_id, TO_CHAR(hire_date, 'MM/YY') "MES AÑO"
FROM employees
WHERE TO_CHAR (hire_date, 'MON') LIKE 'ENE';
```

EMPLOYEE_ID	MES AÑO
102	01/01
103	01/06
127	01/07
142	01/05

```
SELECT employee_id, TO_CHAR(hire_date, 'fmMM/YY') "MES AÑO"
FROM employees
WHERE TO_CHAR (hire_date, 'MON') LIKE 'ENE';
```

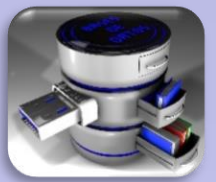
fm elimina els espais en blanc o suprimir els zeros inicials

EMPLOYEE_ID	MES AÑO
102	1/1
103	1/6
127	1/7
142	1/5



Funcion TO_CHAR

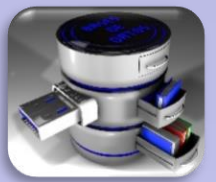
Elemento	Resultado
YYYY	Año completo en números
YEAR	Año en letra (en inglés)
MM	Valor de dos dígitos del mes
MONTH	Nombre completo del mes
MON	Abreviatura de tres letras del mes
DY	Abreviatura de tres letras del día de la semana
DAY	Nombre completo del día de la semana
DD	Día numérico del mes



Funcion TO_CHAR

Altres formats

/ , .	**La puntuació es reprodueix en el resultat**
“texto”	La cadena entre cometes dobles es reprodueix en el resultat
TH	número ordinal (4TH)
SP	Número en lletra (QUATRE)
SPTH oTHSP	Número ordinal en lletra (QUART)



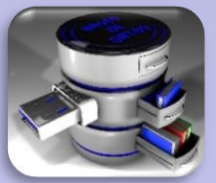
Funcion TO_CHAR

```
SELECT employee_id, TO_CHAR(hire_date, 'DD "de" MM "de" YYYY') fecha
FROM employees;
```

EMPLOYEE_ID	FECHA
100	17 de 06 de 2003
101	21 de 09 de 2005

```
SELECT employee_id, TO_CHAR(hire_date, 'DDTH "de" MM "de" YYYY') fecha
FROM employees;
```

EMPLOYEE_ID	FECHA
100	17TH de 06 de 2003
101	21ST de 09 de 2005



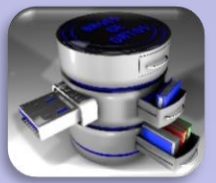
Funcion TO_CHAR

```
SELECT employee_id, TO_CHAR(hire_date, 'DdSP "de" MM "de" YYYY') fecha
FROM employees;
```

EMPLOYEE_ID	FECHA
100	Seventeen de 06 de 2003
101	Twenty-One de 09 de 2005

```
SELECT employee_id, TO_CHAR(hire_date, 'DDSPth "de" MM "de" YYYY') fecha
FROM employees;
```

EMPLOYEE_ID	FECHA
100	SEVENTEENTH de 06 de 2003
101	TWENTY-FIRST de 09 de 2005

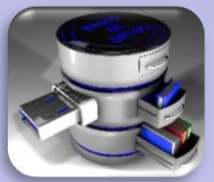


Funcion TO_CHAR

La funció TO_CHAR, també pot convertir números a cadenes de caràcters.
Converteix un valor de tipus NUMBER a un de tipus VARCHAR2

```
TO_CHAR(number, 'format_model')
```

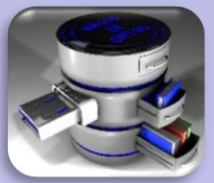
Elemento	Resultado
9	Representa un número
0	Fuerza para que aparezca un cero
$	Coloca un signo de dólar flotante
L	Utiliza el símbolo de divisa local flotante
.	Imprime un punto decimal
,	Imprime una coma como indicador de miles



Funcion TO_CHAR

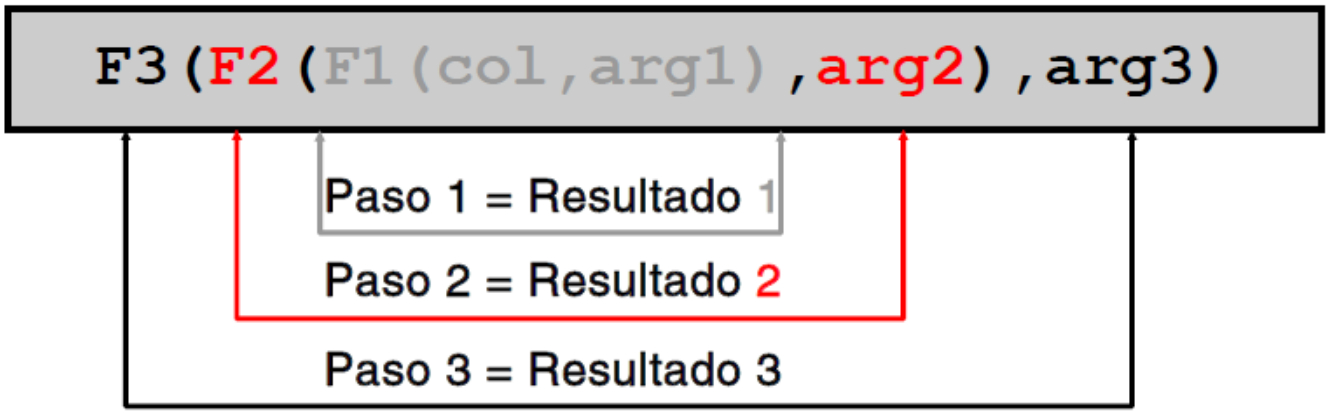
```
SELECT TO_CHAR(salary, '$99,999.00') salario
FROM employees;
```

SALARIO
$24,000.00
$17,000.00
$17,000.00
$9,000.00
$6.000.00



Funcions d'anidació

Les funcions d'una sola fila es poden niar en qualsevol profunditat. Les funcions niades s'avaluen des del nivell més profund fins al nivell menys profund.

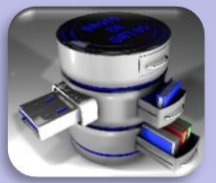


Funcions d'anidació

```
SELECT last_name, UPPER(CONCAT(SUBSTR(last_name, 1, 8),'_US'))
FROM employees;
```

LAST_NAME	UPPER(CONCAT(SUBSTR(LAST_NAME,1,8),'_US'))
Abel	ABEL_US
Ande	ANDE_US
Atkinson	ATKINSON_US

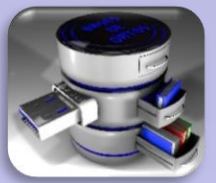
- ✓ La funció interna recupera els huit primers caràcters del cognom.
- ✓ La funció externa concatena el resultat amb _US
- ✓ La funció més externa converteix els resultats a majúscules



Funcions generals

Les següents funcions funcionen amb qualsevol mena de dada i pertanyen a l'ús de valors nuls

- NVL (expr1, expr2)
- NVL2 (expr1, expr2, expr3)
- NULLIF (expr1,expr2)
- COALESCE (expre1, expr2, …, exprn)



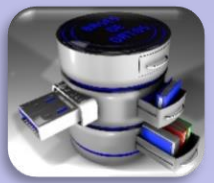
Funcions generals

NVL

- Converteix un valor nul en un valor real
- Els tipus de dades que poden utilitzar-se son: data, carácter i número.
- Els tipus de dades dels arguments han de coincidir.

```
SELECT last_name, salary, NVL(commission_pct, 0)
FROM employees;
```

LAST_NAME	SALARY	NVL(COMMISSION_PCT,0)
Davies	3100	0
Matos	2600	0
Vargas	2500	0
Russell	14000	0,4
Partners	13500	0,3



Funcions generals

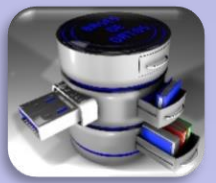
NVL

```
SELECT last_name, salary, (salary + salary *12 *NVL(commission_pct, 0)) "Salario anual"
FROM employees;
```

LAST_NAME	SALARY	Salario anual
Davies	3100	3100
Matos	2600	2600
Vargas	2500	2500
Russell	14000	81200
Partners	13500	62100

```
SELECT last_name, salary, (salary + salary *12 *commission_pct) "Salario anual"
FROM employees;
```

LAST_NAME	SALARY	Salario anual
Davies	3100	(null)
Matos	2600	(null)
Vargas	2500	(null)
Russell	14000	81200
Partners	13500	62100



Funcions generals

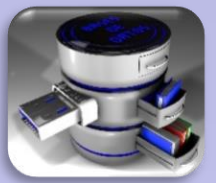
NVL2

Si expr1 no és nul, NVL2 retorna expr2. Si expr1 és nul, NVL2 retorna expr3. L'argument expr1 pot tindre qualsevol tipus de dada

- Els tipus de dades dels arguments han de coincidir.

```
SELECT last_name, salary, NVL2(commission_pct, 'SAL + COM', 'SAL') "Nómina"
FROM employees;
```

LAST_NAME	SALARY	Nómina
Davies	3100	SAL
Matos	2600	SAL
Vargas	2500	SAL
Russell	14000	SAL + COM
Partners	13500	SAL + COM



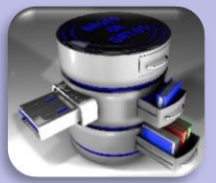
Funcions generals

NULLIF

Compara dues expressions i retorna un valor nul si són iguals; si no són iguals, retorna la primera expressió.

```
SELECT last_name, LENGTH(last_name), first_name, LENGTH(first_name),
NULLIF(LENGTH(last_name),LENGTH(first_name)) COMPARA
FROM employees;
```

LAST_NAME	LENGTH(LAST_NAME)	FIRST_NAME	LENGTH(FIRST_NAME)	COMPARA
Cambrault	9	Gerald	6	9
Cambrault	9	Nanette	7	9
Chen	4	John	4	(null)
Chung	5	Kelly	5	(null)
Colmenares	10	Karen	5	10
Davies	6	Curtis	6	(null)
De Haan	7	Lex	3	7
Dellinger	9	Julia	5	9
Dilly	5	Jennifer	8	5



Funcions generals

COALESCE

Retorna la primera expressió no nul·la en la llista d'expressions

```
SELECT last_name,
COALESCE(TO_CHAR(commission_pct), TO_CHAR(manager_id), 'Sin comisión ni manager')
FROM employees;
```

LAST_NAME	COALESCE(TO_CHAR(COMMISSION_PCT),TO_CHAR(MANAGER_ID),'SINCOMISIÓNNIMANAGER')
King	Sin comisión ni manager
Kochhar	100
De Haan	100
Cambrault	,3
Zlotkey	,2