

CONSULTA DE DADES.

DML (II)



CFGS DAM



1. TIPUS DE SENTÈNCIES D' SQL

- Recordem:
- **DDL**: CREATE, DROP i ALTER
- **DML**: **SELECT**, INSERT, UPDATE i DELETE
- **DCL**: GRANT, REVOKE (concedir, suprimir privilegis) i COMMIT, ROLLBACK (transaccions)

En aquesta secció, aprendreu com mostrar dades d'una taula. En concret, podrem seleccionar les columnes que es mostraran i en l'ordre en què ho faran. Simplement és la instrucció que la base de dades interpreta com sol·licitar informació.

2. CONSULTA DE DADES SIMPLES

- Començarem veient les consultes simples, basades en una taula. Veurem com obtenir files i columnes d'una taula a l'ordre en què ens calga.
- El resultat d'una consulta SELECT ens torna una **taula lògica**.
 - És a dir, els resultats són una relació de dades, que té files/registres, amb una sèrie de camps/columnes. (igual que qualsevol taula de la base de dades).
 - Però aquesta taula està en memòria mentre la utilitzem, i després es descarta.
 - Cada vegada que executem la consulta es torna a calcular el resultat.

2. CONSULTA DE DADES SIMPLES

- Per recuperar alguns camps d'una taula utilitzem la sentència **SELECT**.
- La sintaxi bàsica d'una consulta SELECT és la següent (els valors opcionals van entre claudàtors):

SELECT [**ALL** / **DISTINCT**] [*]
/[LlistaColumnes_Expressions] **AS** [Àlies]

L'asterisc (*) indica que es seleccionen tots els camps de la taula.

FROM Nom_Taula_o_Vista

WHERE Condicions

ORDER BY LlistaColumnes [**ASC** / **DESC**]

Àlies de columnes:
per assignar noms temporals a columnes o expressions en una consulta.

ORDER BY :per ordenar el conjunt de resultats retornats d'una consulta.

2. CONSULTA DE DADES SIMPLES

ALL / DISTINCT

- **ALL** és el valor per defecte, especifica que el conjunt de resultats pot incloure files duplicades. Per regla general mai no s'utilitza.
- **DISTINCT** especifica que el conjunt de resultats només pot incloure files úniques. És a dir, si en fer una consulta hi ha registres exactament iguals que apareixen més d'una vegada, s'eliminen. Molt útil moltes vegades. → Elimina files duplicades.

3. FILTRAT DE DADES: WHERE

- **WHERE:** Especifica la condició de filtre de les files retornades.
- S'utilitza quan no es vol que es tornen totes les files d'una taula, sinó només les que compleixen certes condicions.
- El més habitual és utilitzar aquesta clàusula a la majoria de les consultes.
- **NOTA:** Si utilitzeu àlies de columna a la clàusula SELECT, els podeu utilitzar en la clàusula ORDER BY, però no els podeu utilitzar a la clàusula WHERE.

3. FILTRAT DE DADES: WHERE

- **Condicions:** Són **expressions lògiques** a comprovar per a la condició de filtre, que després de la seua comprovació retornen per a cada fila TRUE o FALSE, en funció que es complisquen o no.
- Es pot utilitzar qualsevol expressió lògica i utilitzar diversos operadors com:
 - > (Major)
 - >= (Major o igual)
 - < (Menor)
 - <= (Menor o igual)
 - = (Igual)
 - <> o != (Distint)

3. FILTRAT DE DADES: WHERE

- **Condicions:**

- IS [NOT] NULL (per comprovar si el valor d'una columna és o no és nul·la, és a dir, si conté o no conté valor).
- **Important!** Es diu que una columna d'una fila és **NULL** si està **completament buida**. Si s'ha introduït qualsevol dada, fins i tot en un camp alfanumèric si s'introdueix una cadena en blanc o un zero en un camp numèric deixa de ser NULL.

3. FILTRAT DE DADES: WHERE

- **Condicions:**

- **LIKE:** per comparar un model. Per això utilitza els caràcters comodí especials: “%” i “_”. Amb el primer indiquem que en lloc seu pot anar qualsevol cadena de caràcters, i amb el segon que pot anar qualsevol caràcter individual (un sol caràcter). Amb la combinació d'aquests caràcters podrem obtenir múltiples patrons de cerca. Per exemple:
 - El nom comença per A: Nom LIKE 'A%'
 - El nom acaba per A: Nom LIKE '%A'
 - El nom conté la lletra A: Nom LIKE '%A%'
 - El nom comença per A i després conté un sol caràcter qualsevol: Nom LIKE 'A_'
 - El nom comença una A, després qualsevol caràcter, després una E i al final qualsevol cadena de caràcters: Nom LIKE 'A_E%'

3. FILTRAT DE DADES: WHERE

- **Condicions:**

- **BETWEEN:** per a un interval de valors. Per exemple:
- Clients entre el 30 i el 100: `CodCliente BETWEEN 30 AND 100`
- Clients nascuts entre 1970 i 1979: `DataNac BETWEEN '19700101' AND '19791231'`
-
- **IN():** per especificar una relació de valors concrets.
- Per exemple: Vendes dels Clients 10, 15, 30 i 75: `CodCliente IN(10, 15, 30, 75)`

3. FILTRAT DE DADES: WHERE

- És possible **combinar** diverses **condicions** simples dels operadors anteriors utilitzant els operadors lògics **OR**, **AND** i **NOT**, així com l'ús de parèntesis per controlar la prioritat dels operadors (com en matemàtiques).
- Per exemple: ... (Client = 100 AND Província = 30) OR Vendes > 1000 ... que seria per al client amb codi 100 de la província 30 o qualsevol client les vendes del qual superen 1000.

4. LIMITAR ELS RESULTATS D'UNA CONSULTA

- També podem **limitar el nº de files** que retorna una consulta.
- **LIMIT** és una clàusula opcional de la sentència SELECT.
- La sintaxi bàsica de la clàusula LIMIT:
- **SELECT** llista_columnes
- **FROM** nom_taula
- **WHERE** Condicions
- **ORDER BY** expressio
- **LIMIT** nombre_files_retorn [**OFFSET** files_a_ignorar];

Amb OFFSET s'ometran les primeres "files_a_ignorar" files abans de tornar les "nombre_files" files generades per la consulta. Si "files_a_ignorar" és 0, la sentència funcionarà igual com si no tinguérem la clàusula OFFSET.

4. LIMITAR ELS RESULTATS D'UNA CONSULTA

- **NOTA:** quan utilitzeu la clàusula LIMIT, sempre heu d'utilitzar la clàusula ORDER BY per controlar l'ordre de les files.
- Si no la utilitzeu, és possible que obtingueu un conjunt de resultats amb l'ordre de files no especificat.

4. LIMITAR ELS RESULTATS D'UNA CONSULTA

- La clàusula LIMIT no és un estàndard SQL.
- Per complir amb l'estàndard SQL, PostgreSQL admet la clàusula FETCH per recuperar un n° de files retornades per una consulta.
- La sintaxi bàsica de la clàusula FETCH de PostgreSQL:
- **SELECT** llista_columnes
- **FROM** nom_taula
- **WHERE** Condicions
- **OFFSET** inici { **ROW** | **ROWS** }
- **FETCH**{ **FIRST** | **NEXT** } [num_files]
{ **ROW** | **ROWS** } **ONLY**

ROW és sinònim de ROWS i FIRST és sinònim de NEXT. Podeu utilitzar-los indistintament

4. LIMITAR ELS RESULTATS D'UNA CONSULTA

- **inici** és un n° enter que ha de ser ≥ 0 . Per defecte, és zero si no s'especifica la clàusula OFFSET.
 - En cas que **inici** siga superior al n° de files del conjunt de resultats, no es retornarà cap fila;
- **num_files** és un n° enter que ha de ser 1 o superior. El valor per defecte de **num_files** és 1 si no ho especifiquem explícitament.