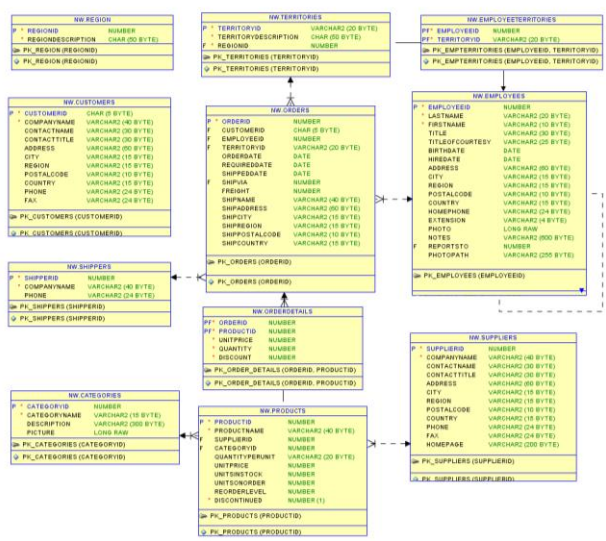
Ripasso linguaggio SQL

Prof. Alessandra Lumini
alessandra.lumini@unibo.it

Schema DB



L'istruzione SELECT

- È l'istruzione che permette di eseguire interrogazioni (*query*) sul DB

```
SELECT  A1,A2,..,Am
FROM    R1,R2,..,Rn
[WHERE   <condizione>]
[GROUP BY <listaAttributi>]
[HAVING <condizione>]
[ORDER BY <listaAttributi>]
```

  - ovvero:
    - SELECT (o TARGET) list (cosa si vuole come risultato)
    - clausola FROM (da dove si prende)
    - clausola WHERE (che condizioni deve soddisfare)
    - clausola GROUP BY (le colonne su cui raggruppare)
    - clausola HAVING (condizione relative ai gruppi)
    - clausola ORDER BY (ordinamento)

Il comando SELECT permette di realizzare le operazioni di selezione, proiezione, join, raggruppamento e ordinamento.

Selezione semplice

- Selezionare tutti i prodotti di categoria 'Beverages'

```
SELECT P.PRODUCTNAME
FROM NW.CATEGORIES C, NW.PRODUCTS P
WHERE C.CATEGORYID=P.CATEGORYID
AND C.CATEGORYNAME='Beverages';

SELECT P.PRODUCTNAME
FROM NW.CATEGORIES C INNER JOIN
NW.PRODUCTS P ON
C.CATEGORYID=P.CATEGORYID
WHERE C.CATEGORYNAME='Beverages';
```

1. Selezionare gli ordini gestiti dall'impiegata Davolio Nancy

Query aggregate

- Contare i prodotti di categoria ‘Beverages’

```
SELECT count(*)
FROM NW.CATEGORIES C, NW.PRODUCTS P
WHERE C.CATEGORYID=P.CATEGORYID
AND C.CATEGORYNAME='Beverages';
```

2. Contare gli ordini gestiti dall’impiegata Davolio Nancy

Query con raggruppamento

- Contare i prodotti di ciascuna categoria

```
SELECT C.CATEGORYNAME, count(*)
FROM NW.CATEGORIES C, NW.PRODUCTS P
WHERE C.CATEGORYID=P.CATEGORYID
group by C.CATEGORYNAME
order by 1;
```

3. Contare gli ordini gestiti da ciascun impiegato

Filtro 'Positivo'

- Selezionare tutti i prodotti presenti in almeno un ordine

```
SELECT PRODUCTNAME
FROM NW.PRODUCTS
where PRODUCTID in (SELECT PRODUCTID FROM
NW.ORDERDETAILS);

SELECT DISTINCT PRODUCTNAME
FROM NW.PRODUCTS P, NW.ORDERDETAILS O
where P.PRODUCTID = O.PRODUCTID;
```

4. Selezionare gli impiegati che hanno gestito almeno un ordine spedito in 'Norway'

7

Filtro 'Negativo' = Sottrazione

- Selezionare tutti i prodotti che non sono presenti in nessun ordine spedito negli 'USA'

```
SELECT PRODUCTNAME
FROM NW.PRODUCTS
where PRODUCTID NOT IN (SELECT PRODUCTID
FROM NW.ORDERDETAILS OD join NW.ORDERS O
on (OD.ORDERID=O.ORDERID) WHERE
SHIPCOUNTRY='USA' );

--errata
SELECT DISTINCT PRODUCTNAME
FROM NW.PRODUCTS P, NW.ORDERDETAILS OD,
NW.ORDERS O
where OD.PRODUCTID = P.PRODUCTID and
OD.ORDERID=O.ORDERID and
SHIPCOUNTRY<>'USA';
```

5. Selezionare gli impiegati che non hanno gestito ordini spediti in 'Norway'

8

Filtri sui gruppi

- Selezionare gli impiegati che hanno gestito più di 50 ordini

```
SELECT A.LASTNAME, A.FIRSTNAME, count(*) as NumeroOrdini
FROM NW.ORDERS B, NW.EMPLOYEES A
WHERE A.EMPLOYEEID = B.EMPLOYEEID
group by A.EMPLOYEEID, A.LASTNAME, A.FIRSTNAME
having count(*)> 50
order by 3 desc;
```

6. Selezionare i prodotti che sono stati venduti a più di 10 clienti diversi

9

Filtro 'Universale' = Divisione

- Selezionare gli impiegati che hanno gestito ordini per tutti i paesi

```
--non ci sono paesi in cui non sia stato spedito un ordine gestito da quell'impiegato
SELECT A.LASTNAME, A.FIRSTNAME
FROM NW.EMPLOYEES A
WHERE NOT EXISTS
  (SELECT * FROM NW.ORDERS O WHERE NOT EXISTS
   (SELECT * FROM NW.ORDERS O1 WHERE A.EMPLOYEEID = O1.EMPLOYEEID AND
O.SHIPCOUNTRY=O1.SHIPCOUNTRY));

--numero paesi serviti= numero paesi presenti nel DB
SELECT A.LASTNAME, A.FIRSTNAME, count(distinct SHIPCOUNTRY) as PaesiDistinti
FROM NW.ORDERS B, NW.EMPLOYEES A
WHERE A.EMPLOYEEID = B.EMPLOYEEID
group by A.EMPLOYEEID, A.LASTNAME, A.FIRSTNAME
having count(distinct SHIPCOUNTRY)=
            (SELECT count(distinct SHIPCOUNTRY) FROM NW.ORDERS O);
```

7. Selezionare clienti che sono stati serviti da tutti i corrieri

10

Self-join

- Selezionare le coppie di impiegati della stessa città (senza duplicati)

```
SELECT e1.EMPLOYEEID,e2.EMPLOYEEID,e1.CITY
FROM NW.EMPLOYEES e1,NW.EMPLOYEES e2
where e1.EMPLOYEEID<e2.EMPLOYEEID and e1.CITY=e2.CITY;
```

- Selezionare gli impiegati che hanno un superiore che abita nella stessa città

```
SELECT e1.EMPLOYEEID as Impiegato ,e2.EMPLOYEEID as Superiore, e1.CITY
FROM NW.EMPLOYEES e1,NW.EMPLOYEES e2
where e1.REPORTSTO =e2.EMPLOYEEID and e1.CITY=e2.CITY;
```

8. Selezionare le coppie di clienti della stessa nazione (senza duplicati)

9. Selezionare gli impiegati che hanno un superiore della stessa età (anno di nascita)

Outer-join

- Selezionare gli impiegati e il relativo superiore, inclusi gli impiegati che non hanno superiore

```
SELECT e.EMPLOYEEID, e.LASTNAME, e.FIRSTNAME, e.REPORTSTO as Sup,
s.LASTNAME as cognomeSup, s.FIRSTNAME as nomeSup
FROM NW.EMPLOYEES e left outer join NW.EMPLOYEES s on (
e.REPORTSTO=s.EMPLOYEEID);
```

10. Ordinare i clienti in base al numero di ordini effettuati nel 1996 (includere anche i clienti che non hanno fatto ordini)

Max(Count())

- Selezionare l'impiegato che ha gestito il maggior numero di ordini

```
SELECT e.EMPLOYEEID, e.LASTNAME, e.FIRSTNAME,
count(ORDERID) as NumOrdini
FROM NW.EMPLOYEES E join  NW.ORDERS o on
(e.EMPLOYEEID=o.EMPLOYEEID)
group by e.EMPLOYEEID, e.LASTNAME, e.FIRSTNAME
having count(ORDERID)>=ALL (SELECT  count(ORDERID) as
NumOrdini
FROM  NW.ORDERS
group by EMPLOYEEID);
```

11. Selezionare il cliente che ha effettuato il maggior numero di ordini
12. Selezionare il cliente che ha generato il più alto introito nel 1996