

**Laboratorio di Basi di Dati**                      **Matricola:**  
**Appello del 04/07/2019** (100 minuti) **Nome e cognome:**  
Login: esame\_\_\_\_

FILA

A

1) Il sistema informativo di una compagnia di taxi mantiene le informazioni relative ai taxi, i conducenti e alle corse.

TAXI(T\_numTaxi, T\_numeroPosti, T\_codDriver:DRIVER)

DRIVER(D\_codDriver, D\_nome, D\_cognome, D\_dataNascita)

CORSE(C\_numTaxi:TAXI, C\_daIndirizzo, C\_aIndirizzo, C\_oraPartenza, C\_oraArrivo, C\_km, C\_costo)

Si vuole calcolare una statistica sulle fasce orarie con maggiore mole di lavoro e salvare i dati nella tabella:

OUTPUT(O\_oraInizio, O\_oraFine, O\_numCorse, O\_kmMedi, O\_costoMedio)

a) Si definisca la base di dati su ORACLE (facendo attenzione al Dominio utilizzato, es. INT per gli ID, TIMESTAMP per le ore, FLOAT per costi e km).

b) Si scriva la funzione FasceOrarie (vNumOre) che ripulisce la tabella output e inserisce i dati aggregati relativi alla suddivisione della giornata in fasce orarie di ampiezza vNumOre, infine stampa a video le 5 fasce con maggior numero di corse effettuate.

**Esempio** Si supponga che il parametro vNumOre =2 si dovranno calcolare il numero di corse, costo medio e km medi percorsi nelle fasce orarie dalle 0-2, 2-4, 4-6, ... e stampare le 5 fasce con più richieste:

Fascia da 8 a 10 corse: 7

Fascia da 10 a 12 corse: 4

Fascia da 14 a 16 corse: 3

Fascia da 12 a 14 corse: 2

Fascia da 6 a 8 corse: 1

**ATTENZIONE: Nel caso in cui i campi Nome, Cognome e Login non siano compilati in modo leggibile il compito non sarà corretto**

```
create table A20190704_TAXI (  
T_numTaxi int,  
T_numeroPosti int,  
T_codDriver int,  
primary key (T_numTaxi),  
foreign key (T_codDriver) references A20190704_DRIVER(D_codDriver)  
);
```

```
create table A20190704_DRIVER(  
D_codDriver int,  
D_nome varchar2(16),  
D_cognome varchar2(16),  
D_datanascita DATE,  
primary key(D_codDriver)  
);
```

```
create table A20190704_CORSE(  
C_numTaxi int,  
C_indirizzoDa varchar2(16),  
C_indirizzoA varchar2(16),  
C_oraPartenza timestamp,  
C_oraArrivo timestamp,  
C_km float,  
C_costo float,  
primary key(C_numTaxi,C_oraPartenza),  
foreign key (C_numTaxi) references A20190704_TAXI(T_numTaxi)  
);
```

```
create table A20190704_OUTPUT(  
O_oraInizio int,  
O_oraFine int,  
O_numCorse int,  
O_kmMedi float,  
O_costoMedio float,  
primary key(O_oraInizio,O_oraFine)  
);
```

```

create or replace procedure A20190704_FasceOrarie(vNumOre INT) IS
--cursore
cursor cOut is
select * from A20190704_OUTPUT
order by O_numCorse DESC;

vOut cOUT%ROWTYPE;
vCont int;

begin
--ripulisce la tabella di output
delete from A20190704_OUTPUT;
--carica i dati
insert into A20190704_OUTPUT select TRUNC(to_number(EXTRACT(hour FROM
c_oraPartenza))/vNumOre)*vNumOre as oraInizio, max(TRUNC(to_number(EXTRACT(hour FROM
c_oraPartenza))/vNumOre)*vNumOre+vNumOre)as oraFine, count(*) as C_numCorse,avg(C_km) as
kmMedi,avg(c_costo) AS costoMedio
from A20190704_CORSE
GROUP BY TRUNC(to_number(EXTRACT(hour FROM c_oraPartenza))/vNumOre)*vNumOre;

--stampa
open cOut;
vCont:=0;
loop
--recupero di ogni PR
fetch cOut into vOut;
vCont:=vCont+1;
exit when cOut%NOTFOUND;

--stampa
DBMS_OUTPUT.PUT_LINE('Fascia da ' || vOut.O_oraInizio || ' a ' || vOut.O_oraFine || '
corse: ' || vOut.O_numCorse);
exit when vCont>=5;

end loop;
close cOut;
end;

```

```

select  P_BRAND, sum(L_QUANTITY)
from    TPCD.PART,TPCD.LINEITEM,TPCD.ORDERS
where   P_PARTKEY=L_PARTKEY and L_ORDERKEY=O_ORDERKEY
        and P_CONTAINER='WRAP CAN' and O_ORDERSTATUS='F'
group by P_BRAND;

```

OPERATION	OBJECT_NAME	CARDINALITY	C
SELECT STATEMENT			25
SORT (GROUP BY)			25
NESTED LOOPS			50380
NESTED LOOPS			151141
TABLE ACCESS (FULL)	PART		5037
Filter Predicates			
P_CONTAINER='WRAP CAN'			
TABLE ACCESS (BY INDEX ROWID)	LINEITEM		30
INDEX (RANGE SCAN)	IX_PART_LI		30
Access Predicates			
P_PARTKEY=L_PARTKEY			
TABLE ACCESS (BY INDEX ROWID)	ORDERS		1
Filter Predicates			
O_ORDERSTATUS='F'			
INDEX (UNIQUE SCAN)	SYS_C0036431		1
Access Predicates			
L_ORDERKEY=O_ORDERKEY			

$$NP_{PART} = \lceil 200,000 \times 131 / (4096 \times 0,69) \rceil = 9,271$$

$$NP_{LINEITEM} = \lceil 6,001,215 \times 116 / (4096 \times 0,69) \rceil = 246,314$$

$$NP_{ORDERS} = \lceil 1,500,000 \times 106 / (4096 \times 0,69) \rceil = 56,259$$

$$Sel_{(P\_CONTAINER=...)} = 1/40$$

$$Sel_{(O\_ORDERSTATUS=...)} = 1/3$$

$$NL_{(L\_PARTKEY)} = \lceil (200,000 \times 4 + 6,001,215 \times 4) / (4096 \times 0,69) \rceil = 8,777$$

$$\text{Accesso a LI} = 2 + \lceil 1/200,000 \times 8,777 \rceil + \Phi(6,001,215/200,000; 246,314) = 34$$

$$\text{Costo Join Part+Lineitem} = 9,271 + 1/40 \times 200,000 \times 34 = \mathbf{179,271}$$

$$NP_{P-LI} = \lceil 6,001,215 \times 1/40 \rceil = 150,031$$

$$NP_{P-LI} = \lceil 24,005 \times (116+131) / (4096 \times 0,69) \rceil = 13,112$$

$$NL_{(O\_ORDERKEY)} = \lceil (1,500,000 \times 4 + 1,500,000 \times 4) / (4096 \times 0,69) \rceil = 4,246$$

$$\text{Accesso a ORDERS} = 2 + \lceil 1/1,500,000 \times 4,246 \rceil + \lceil 56,259/1,500,000 \rceil = 4$$

$$\text{Costo Join P+LI+O} = 13,112 + 150,031 \times 4 = \mathbf{613,236}$$

$$NP_{P-LI} = \lceil 6,001,215 \times 1/40 \times 1/3 \rceil = 50,011$$

$$NP_{P-LI} = \lceil 50,011 \times (106+116+131) / (4096 \times 0,69) \rceil = 6,247$$

$$\text{Costo del group by} = 2 \times 6,247 \times (\lceil \log_{100} 6,247 \rceil + 1) = \mathbf{37,482}$$

$$\mathbf{\text{Costo Totale} = 179,271 + 613,236 + 37,482 = 829,989}$$