Povezani upiti

- Prikazati mbr, ime, prz, plt radnika čiji je broj sati angažovanja na nekom projektu veći od prosečnog broja sati angažovanja na tom projektu.

```
select distinct r.mbr, ime, prz, plt, brc
from radnik r, radproj rp1
where r.mbr=rp1.mbr and
rp1.brc>(select avg(brc) from radproj rp2
where rp2.spr=rp1.spr);
```

EXISTS

EXISTS(<lista_vrednosti>) – <lista_vrednosti> nije prazan skup vrednosti

NOT EXISTS(<lista_vrednosti>) – <lista_vrednosti> je prazan skup vrednosti

EXISTS

- Ko je najstariji radnik? (exist)

```
select ime, prz, god
from radnik r
where not exists
(select mbr from radnik r1
where r1.god<r.god);
```

EXISTS

- Izlistati radnike koji ne rade ni na jednom projektu. (ne postoji projekat na kom rade)

```
select  mbr, ime, prz
from radnik r
where not exists
(select * from radproj rp where r.mbr=rp.mbr);

select  mbr, ime, prz
from radnik r
where mbr not in
(select rp.mbr from radproj rp);
```

Povezani upiti - EXISTS

- Ko je najmlađi rukovodilac projekata?

```
select distinct mbr, ime, prz, god
from radnik r, projekat p
where r.mbr=p.ruk and not exists
(select mbr from radnik r1, projekat p1
where r1.mbr=p1.ruk and r1.god>r.god);
```

Unija (UNION)

- Izlistati mbr, ime, prz radnika koji rade na projektu sa šifrom 20 ili im je plata veća od prosečne. (unija)

```
select mbr, ime, prz from radnik
where mbr in
(select mbr from radproj where spr=20)
union
select mbr, ime, prz from radnik
where plt>(select avg(plt) from radnik);
```

Unija (UNION ALL)

- Izlistati mbr, ime, prz radnika koji rade na projektu sa šifrom 20 ili im je plata veća od prosečne. (unija)

```
select mbr, ime, prz from radnik
where mbr in
(select mbr from radproj where spr=20)
union all
select mbr, ime, prz from radnik
where plt>(select avg(plt) from radnik);
```

Presek (INTERSECT)

- Izlistati mbr, ime, prz radnika čije prezime počinje na slovo M ili slovo R i mbr, ime, prz radnika čije prezime počinje na slovo M ili slovo P.

```
select mbr, ime, prz from radnik
where prz like 'M%' or prz like 'R%'
INTERSECT
select mbr, ime, prz from radnik
where prz like 'M%' or prz like 'P%';
```

Razlika (MINUS)

- Izlistati mbr, ime, prz radnika čije prezime počinje na slovo M ili slovo R i mbr, ime, prz radnika čije prezime počinje na slovo M ili slovo P.

```
select mbr, ime, prz from radnik
where prz like 'M%' or prz like 'R%'
MINUS
select mbr, ime, prz from radnik
where prz like 'M%' or prz like 'P%';
```

Prirodno spajanje (NATURAL)

- Prikazati ime i prz radnika koji rade na projektu sa šifrom 30.

select ime, prz
from radnik natural join radproj
where spr=30;

Spajanje se vrši na osnovu imena kolona.

Unutrašnje spajanje (INNER)

- Prikazati ime i prz radnika koji rade na projektu sa šifrom 30.

```
select ime, prz
from radnik r inner join radproj rp
on r.mbr=rp.mbr
 where spr=30;
```

Spoljno spajanje (OUTER)

- Levo (LEFT)
- Desno (RIGHT)
- Potpuno (FULL)

Spoljno spajanje (LEFT OUTER)

- Prikazati mbr, ime i prz svih radnika i nazive projekata kojima rukovode. Ukoliko radnik ne rukovodi ni jednim projektom ispisati: ne rukovodi projektom.

select r.mbr,ime, prz, nvl(nap, 'ne rukovodi projektom') Projekat

from radnik r left outer join projekat p

on r.mbr=p.ruk;

Spoljno spajanje (RIGHT OUTER)

- Prikazati nazive svih projekata i mbr radnika koji rade na njima. Ukoliko na projektu ne radi ni jedan radnik ispisati nulu umesto matičnog broja.

```
select nvl(rp.mbr, 0) "Mbr radnika", nap
from radproj rp right outer join projekat p
on rp.spr=p.spr;
```

```
select nvl(rp.mbr, 0) "Mbr radnika", nap
from radproj rp,  projekat p
where rp.spr(+)=p.spr;
```

Spoljno spajanje (FULL OUTER)

```
select nvl(rp.mbr, 0) "Mbr radnika", nap
from radproj rp full outer join projekat p
on rp.spr=p.spr;
```

Primer

- Prikazati za sve radnike i projekte na kojima rade Mbr, Prz, Ime, Spr i Nap. Za radnike koje ne rade ni na jednom projektu, treba prikazati Mbr, Prz i Ime, dok za vrednosti obeležja Spr i Nap treba zadati, redom, konstante 0 i "Ne postoji". Urediti izlazni rezultat saglasno rastućim vrednostima obeležja Mbr.

Rešenje

```
SELECT r.Mbr, r.Prz, r.Ime, NVL(p.Spr, 0) AS Spr,
NVL(p.Nap, 'Ne postoji') AS Nap
FROM Radnik r, Radproj rp, Projekat p
WHERE r.Mbr = rp.Mbr (+) AND rp.Spr = p.Spr (+)
ORDER BY Mbr;
```

```
SELECT r.Mbr, r.Prz, r.Ime, NVL(p.Spr, 0) AS Spr,
NVL(p.Nap, 'Ne postoji') AS Nap
FROM Radnik r LEFT OUTER JOIN Radproj rp ON
r.Mbr = rp.Mbr LEFT OUTER JOIN Projekat p ON
rp.Spr = p.Spr
ORDER BY Mbr;
```

Dekartov proizvod spajanje (Cross Join)

- Koristi se ako želimo da napravimo Dekartov proizvod između dve tabele

  SELECT * FROM radnik, projekat

- Ekvivalentno je sa

  **SELECT *
  FROM radnik CROSS JOIN projekat;**

- Može se dodati uslov na cross join, onda se ponaša kao inner join
- Često se zaborave uslovi spoja prilikom spajanja tabela, pa rezultat bude Dekartov proizvod torki iz spajajućih tabela

Selekcioni izraz (CASE)

- Prosti (*Simple*) CASE:

```
CASE expr
        WHEN expr1 THEN return_expr1
        [ WHEN expr2 THEN return_expr2
        WHEN exprn THEN return_exprn]
        [ ELSE else_expr ]
END;
```

- Pretražujući (*Searched*) CASE:

```
CASE
        WHEN comparison_expr1 THEN return_expr1
        [ WHEN comparison_expr2 THEN return_expr2
        WHEN comparison_exprn THEN return_exprn]
        [ ELSE else_expr ]
END;
```

Primer

- Za svakog radnika prikazati mbr, ime, prz, kao kategoriju kojoj pripada na osnovu visine plate. Kategorije po visini plate su sledeće:
  - Plata manja od 10000 – kategorija: **'mala primanja'**,
  - plata između 10000 i 20000 – kategorija: **'srednje visoka primanja'**,
  - plata između 20000 i 40000 – kategorija: **'visoka primanja'**,
  - plata veća od 40000 – kategorija: **'izuzetno visoka primanja'**.
- Takođe, radnike urediti prema kategoriji kojoj pripadaju, u redosledu od najniže ka najvišoj kategoriji po visini plate.

Primer

```
select mbr, ime, plt,
case
   when plt < 10000 then 'mala primanja'
   when plt >=10000 and plt < 20000 then 'srednja primanja'
   when plt >=20000 and plt < 40000 then 'visoka primanja'
   else 'izuzetno visoka primanja'
end as visina_primanja
from radnik
order by
case visina_primanja
   when  'izuzetno visoka primanja' then 1
   when 'visoka primanja' then 2
   when  'srednja primanja' then 3
   else 4
end desc, plt asc;
```

Selekcioni izraz (CASE)

- Napomene:
  - Može se iskoristiti gde god je dozvoljeno korišćenje izraza
    - Najčešće – u okviru liste kolona ili u okviru *order by* klauzule
  - Ukoliko se ne iskoristi *else*, podrazumevana vrednost biće *null*
  - Kod prostog CASE izraza, poređenje sa *null* vrednošću **nije moguće** – podrazumevano se koristi poređenje operatorom '**=**', pa je takav izraz uvek netačan

Primer

- Za svakog radnika ispisati mbr, ime, prz, platu i mbr šefa. Pri ispisu treba obezbediti da su radnici uređeni saglasno visini plate, od najviše ka najnižoj, pri čemu bi direktor firme trebalo da se ispiše prvi.

```
select mbr, ime, plt, sef
from radnik
order by
case
   when sef is null then 1
   else 2
end, plt desc;
```

Ažuriranje baze podataka

- **INSERT**
- **DELETE**
- **UPDATE**

Ažuriranje baze podataka

- INSERT – dodavanje nove torke

INSERT INTO <naziv_tabele>
[(<lista_obeležja)] VALUES
(<lista_konstanti) | SELECT …

Ažuriranje baze podataka

- INSERT – dodavanje nove torke

insert into Radnik (mbr, ime, prz, plt, sef, god) values (201, 'Ana', 'Savic', 30000, null, '18-aug-71');

insert into Projekat (spr, nap, ruk) values (90, 'P1', 201);

insert into RadProj (mbr, spr, brc) values (201, 90, 5);

Ažuriranje baze podataka

- Probati dodavanje nove torke sa postojećim ključem
- Probati dodavanje novog radnika sa vrednosti *null* postavljenom za IME
- Probati dodavanje nove torke sa nedozvoljenom vrednosti PLT (<500)
- Probati dodavanje novog projekta sa postojećim nazivom
- Probati dodavanje novog projekta sa nepostojećim MBR-om rukovodioca

Ažuriranje baze podataka

- DELETE – brisanje postojećih torki

DELETE FROM <naziv_tabele>
[WHERE (<uslov_selekcije>)]

Ažuriranje baze podataka

- DELETE – brisanje postojećih torki

  delete radnik;

  delete radnik where mbr=701;

Ažuriranje baze podataka

- Probati brisanje torke koja je referencirana od strane neke druge torke.

Ažuriranje baze podataka

- UPDATE – modifikacija postojećih torki

UPDATE <naziv_tabele>

SET <obeležje>= <aritm_izraz>
{,<obeležje>= <aritm_izraz>}

[WHERE (<uslov_selekcije>)]

Ažuriranje baze podataka

- UPDATE – modifikacija postojećih torki

```
update radnik
set plt = plt*1.2;

update radnik
set plt = plt*1.2
where mbr = 201;
```

Ažuriranje baze podataka

- Probati ažuriranje torke sa nedozvoljenom vrednosti PLT (<500)