

2016.

```
SELECT datumMec,  
SUBSTRING(igrac1.imeIgrac FROM 1 FOR 1) || '. ' ||  
TRIM(igrac1.prezIgrac) || ' - ' ||  
SUBSTRING(igrac2.imeIgrac FROM 1 FOR 1) || '. ' ||  
TRIM(igrac2.prezIgrac) AS igraci,  
ukSetova1+ukSetova2 AS ukupnoSetova  
FROM mec  
JOIN igrac igrac1 ON igrac1.sifIgrac = mec.sifIgrac1  
JOIN igrac igrac2 ON igrac2.sifIgrac = mec.sifIgrac2  
WHERE (CURRENT_DATE - datumMec) <= 100  
ORDER BY datumMec, ukupnoSetova DESC
```

```
SELECT igrac.sifIgrac, imeIgrac, prezIgrac,  
CASE WHEN SUM(ukGemova1) IS NULL THEN 0  
ELSE SUM(ukGemova1) END,  
CASE WHEN SUM(ukGemova2) IS NULL THEN 0  
ELSE SUM(ukGemova2) END  
FROM set  
JOIN mec  
ON set.datumMec = mec.datumMec  
AND set.sifIgrac1 = mec.sifIgrac1  
JOIN turnir  
ON turnir.sifTurnir = mec.sifTurnir  
AND turnir.ukupnoKola = (SELECT MAX(ukupnoKola) FROM  
turnir)  
RIGHT JOIN igrac  
ON igrac.sifIgrac = mec.sifIgrac1  
WHERE korisnik IS NULL  
GROUP BY igrac.sifIgrac, imeIgrac, prezIgrac
```

```

CREATE TABLE mec(
datumMec DATE,
sifIgrac1 SMALLINT
REFERENCES igrac(sifIgrac),
sifIgrac2 SMALLINT NOT NULL REFERENCES
igrac(sifIgrac),
sifTurnir SMALLINT NOT NULL REFERENCES turnir
(sifTurnir) ON DELETE CASCADE,
ukSetova1 SMALLINT DEFAULT 0 NOT NULL,
ukSetova2 SMALLINT DEFAULT 0 NOT NULL,
CONSTRAINT chkHostGuest CHECK (sifIgrac1 !=
sifIgrac2),
CONSTRAINT chkSetDiff CHECK ((ukSetova1 - ukSetova2)
BETWEEN -2 AND 2),
PRIMARY KEY (datumMec, sifIgrac1));

```

```

CREATE OR REPLACE FUNCTION incSet () RETURNS trigger
AS $$
BEGIN
    UPDATE mec
    SET ukSetova1 = ukSetova1 +1
    WHERE mec.datumMec = NEW.datumMec
    AND mec.sifIgrac1 = NEW.sifIgrac1
    AND NEW.ukGemova1 > NEW.ukGemova2;
    UPDATE mec
    SET ukSetova2 = ukSetova2 +1
    WHERE mec.datumMec = NEW.datumMec
    AND mec.sifIgrac1 = NEW.sifIgrac1
    AND NEW.ukGemova1 < NEW.ukGemova2;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
DROP TRIGGER insSet ON set;
CREATE TRIGGER insSet
BEFORE INSERT ON set

```

```
FOR EACH ROW EXECUTE PROCEDURE incSet();
```

```
CREATE VIEW vMec AS
SELECT * FROM mec
WHERE EXISTS (SELECT *
FROM igrac
WHERE (mec.sifIgrac1 = igrac.sifIgrac
OR mec.sifIgrac2 = igrac.sifIgrac)
AND igrac.korisnik = USER)
WITH CHECK OPTION;

GRANT CONNECT TO ihabus;
GRANT SELECT ON vMec TO ihabus;
GRANT INSERT, UPDATE (datumMec, sifIgrac1, sifIgrac2,
sifTurnir) ON vMec TO ihabus;
```

2015.

```
SELECT pas.imePsa, pasmina.nazPasmina,
izlozbaRezultat.datum, plasman
FROM izlozbaRezultat, pas, pasmina, pasmina
nadPasmina
WHERE izlozbaRezultat.kataloskiBroj =
pas.kataloskiBroj
AND pas.sifPasmina = pasmina.sifPasmina
AND pasmina.sifNadPasmina = nadPasmina.sifPasmina
AND nadPasmina.nazPasmina = 'Jazavčari'
AND izlozbaRezultat.datum BETWEEN CURRENT_DATE-30 AND
CURRENT_DATE;
```

```
SELECT pasmina.nazPasmina, COUNT(DISTINCT
izlozbaRezultat.kataloskiBroj) brojPasa
```

```

FROM izlozbaRezultat
JOIN pas
ON izlozbaRezultat.kataloskiBroj = pas.kataloskiBroj
AND EXTRACT(YEAR FROM izlozbaRezultat.datum) =
EXTRACT(YEAR FROM CURRENT_DATE)
RIGHT JOIN pasmina
ON pas.sifPasmina = pasmina.sifPasmina
WHERE pasmina.nazPasmina LIKE '%ovčar%'
AND pasmina.sifNadPasmina IS NULL
GROUP BY pasmina.sifPasmina, pasmina.nazPasmina
ORDER BY brojPasa DESC, pasmina.nazPasmina;

```

```

CREATE TABLE izlozbarezultat (
datum DATE,
kataloskibroj SMALLINT CONSTRAINT fkizlozbarezpas
REFERENCES pas (kataloskibroj),
ukbod SMALLINT NOT NULL,
plasman SMALLINT CONSTRAINT chkplasman CHECK (plasman
IS NULL OR plasman >= 1),
CONSTRAINT pkizlozbarez PRIMARY KEY (datum,
kataloskibroj),
CONSTRAINT chkplasmanukbod CHECK (ukbod = 0 AND
plasman IS NULL OR ukbod > 0)
);

```

```

CREATE OR REPLACE FUNCTION pasPripadaRazredu (
  p_kataloskiBroj CHAR,
  p_datum DATE
) RETURNS SMALLINT AS $$
BEGIN
RETURN (SELECT razred.sifRazred
FROM pas, razred
WHERE pas.kataloskiBroj = p_kataloskiBroj
AND pas.datRodj + razred.starMjeseciOd || '

```

```

months'::INTERVAL <= p_datum
AND pas.datRodj + razred.starMjeseciDo || '
months'::INTERVAL > p_datum);
END;
$$ LANGUAGE plpgsql;

```

```

CREATE OR REPLACE FUNCTION pogreska() RETURNS trigger
AS $$
BEGIN
    RAISE EXCEPTION 'Pas ne pripada nijednom razredu: %
- % (%)!', NEW.kataloskiBroj, TRIM(NEW.imePsa),
NEW.datRodj USING HINT = 'Provjerite podatke';
    RETURN NULL;
END;
$$ LANGUAGE plpgsql;

```

```

CREATE TRIGGER insIzlozbaRezultat
BEFORE INSERT ON izlozbaRezultat
FOR EACH ROW
WHEN (pasPripadaRazredu (NEW.kataloskiBroj,
NEW.datum) IS NULL
)
EXECUTE PROCEDURE pogreska ();

```

## 2014.

```

SELECT nazAerodrom,
CASE WHEN SUM(tezinaTeret) IS NULL THEN 0
ELSE SUM(tezinaTeret)
END
FROM aerodrom LEFT OUTER JOIN teret
    ON sifAerodrom = sifAerodromDo
    AND (datLet IS NULL OR datLet > CURRENT_DATE)
GROUP BY sifAerodrom, nazAerodrom;

```

```

SELECT avion.sifAvion, nosivost, sifAerodromOd,
sifAerodromDo,
odAerodrom.maxNosivost, doAerodrom.maxNosivost
FROM avion, teret, aerodrom AS odAerodrom, aerodrom
AS doAerodrom
WHERE teret.sifAerodromOd = odAerodrom.sifAerodrom
AND teret.sifAerodromDo = doAerodrom.sifAerodrom
AND avion.nosivost <= odAerodrom.maxNosivost
AND avion.nosivost <= doAerodrom.maxNosivost
AND teret.sifAvion IS NULL
    GROUP BY avion.sifAvion, nosivost, sifAerodromOd,
    sifAerodromDo, odAerodrom.maxNosivost,
doAerodrom.maxNosivost
HAVING SUM (tezinaTeret) <= avion.nosivost;

```

```

UPDATE avion
SET datVrijednost = CURRENT_DATE,
vrijednost = vrijednost * (1 - (SELECT COUNT
(DISTINCT datLet)/1000
FROM teret
WHERE teret.sifAvion = avion.sifAvion
    AND teret.datLet > avion.datVrijednost
    AND teret.datLet < CURRENT_DATE));

```

```

CREATE FUNCTION dojavGresku () RETURNS trigger AS $$
BEGIN
    RAISE EXCEPTION 'Godina izgradnje aviona ne može
biti veća od datuma leta!';
END;
$$ LANGUAGE plpgsql;
CREATE TRIGGER insTeret
BEFORE INSERT ON teret

```

```
FOR EACH ROW
WHEN (NEW.datLet IS NOT NULL AND NEW.sifAvion IS NOT
NULL
AND EXISTS (SELECT * FROM avion WHERE sifAvion =
NEW.sifAvion
AND godIzg > EXTRACT(YEAR FROM NEW.datLet)))
EXECUTE PROCEDURE dojaviGresku();
```

```
ALTER TABLE teret ADD CONSTRAINT chkRazliciteSif
CHECK (sifAerodromOd <> sifAerodromDo);
```

```
GRANT CONNECT TO kolar;
CREATE VIEW teretV AS
SELECT * FROM teret
WHERE sifAerodromOd IN (SELECT sifAerodrom FROM
aerodrom
WHERE oznDrzava = 'HR')
WITH CHECK OPTION;
GRANT SELECT, INSERT, UPDATE ON teretV TO kolar;
GRANT SELECT ON avion TO kolar;
GRANT SELECT ON aerodrom TO kolar;
```