DI- FCT-NOVA **9 de maio de 2018**

Bases de Dados

2º teste, 2017/18

Uma resolução

Grupo 1

1 a)

$$\Pi_{nome}\Big[Pessoas \bowtie \sigma_{localidade='Almada'}(Moradas) \bowtie \Pi_{idPessoa}\big[Contratos \bowtie Assinaturas \bowtie \sigma_{preço>50}(Serviços)\big]\Big]$$

```
SELECT DISTINCT nome
FROM Pessoas INNER JOIN Moradas USING (idMorada)
             INNER JOIN Contratos USING (idPessoa)
             INNER JOIN Assinaturas USING (idContrato)
             INNER JOIN Serviços USING (idServ)
WHERE localidade = 'Almada' AND preço > 50;
```

1 b)

$$\Pi_{idContratato}[Assinaturas \div \Pi_{idServ}(Serviços)]$$

```
SELECT DISTINCT A.idContrato
FROM Assinaturas as A
WHERE NOT EXISTS ( (SELECT idServ FROM SERVIÇOS)
                    EXCEPT
                    (SELECT B.idServ FROM Assinaturas as B
                     WHERE A.idContrato = B.idContrato)
                  )
```

1 c)

$$Pessoas \bowtie \rho_{aux(idPessoa)}\Big\{\Pi_{idFunc}\big\{Contratos \bowtie Assinaturas \bowtie \big[\Pi_{idServ}(Serviços) - [\Pi_{idServ}(ServiçosTV) \cup \Pi_{idServ}(ServiçosInternet)]\big]\big\}\Big\}$$

```
SELECT DISTINCT p.*
FROM Pessoas as p INNER JOIN Contratos c ON (p.idPessoa = c.idFunc)
                  INNER JOIN Assinaturas USING (idContrato)
                  INNER JOIN ( (SELECT idServ FROM SERVIÇOS)
                               EXCEPT
                               (SELECT idServ FROM ServiçosTV as B
                                UNION
                                SELECT idServ FROM ServiçosInternet
                               )
                             )
```

2 a)

```
SELECT idContrato, SUM(preço)
FROM Assinaturas NATURAL INNER JOIN Serviços
GROUP BY idContrato;
```

2 b)

```
WITH faturação_servico (idServ,total) AS
     (SELECT idServ, SUM(preço)
      FROM Assinaturas NATURAL INNER JOIN Serviços
      GROUP BY idServ),
     max_faturação (total) AS
     (SELECT MAX(total) FROM faturação_serviços)
SELECT idServ
FROM faturação_serviço NATURAL INNER JOIN max_faturação;
```

2 c)

A vista devolve as pessoas que efetuaram contratos sem termo (com data de fim a NULL). A vista não pode devolver tuplos repetidos pois a cláusula GROUP BY nunca gera grupos repetidos.

Grupo 2

1 a)

```
ALTER TABLE Serviços ADD CONSTRAINT c1 UNIQUE (nome,preço);
```

1 b)

```
ALTER TABLE Contratos ADD CONSTRAINT c2 CHECK ( idPessoa <> idFunc );
```

1 c)

```
CREATE ASSERTION sem_serviço CHECK
( NOT EXISTS (
      SELECT * FROM Assinaturas a INNER JOIN ServiçosTV USING (idServ)
      WHERE NOT EXISTS (
            SELECT *
            FROM Assinaturas b
            WHERE a.idContrato = b.idContrato AND b.idSev = idServINT
      )
  )
);
```

Nota: ignoram-se problemas provocados por tabelas em mutação (mutating tables)

2 a)

```
CREATE TRIGGER mete_internet
AFTER INSERT ON PacoteServiços
REFERENCING NEW ROW as nrow
FOR EACH ROW
DECLARE ha_gratuito NUMBER;
BEGIN
        INSERT INTO PacoteServiços
        SELECT nrow.idServPac, idServINT, 100
        FROM ServiçosTV LEFT OUTER JOIN PacoteServiços
             ON (idServ = idServInc AND idServPac = nrow.idServPac)
        WHERE idServ = nrow.idServInc AND idServINC IS NULL;
END;
```

2 b)

```
CREATE TRIGGER so_um_gratuito
BEFORE INSERT OR UPDATE OF desconto ON PacoteServiços
REFERENCING NEW ROW as nrow
FOR EACH ROW WHEN (nrow.desconto = 100)
DECLARE ha_gratuito NUMBER;
BEGIN
        SELECT COUNT(*) INTO ha_gratuito
        FROM PacoteServiços
        WHERE idServPac = nrow.idServPac AND desconto = 100;
        IF ha_gratuito > 0 THEN
             nrow.desconto := 10;
        END IF;
END;
```

2 c)

```
CREATE TRIGGER nao_dups
AFTER INSERT OR UPDATE OF idServPac, idServInc ON PacoteServiços
REFERENCING NEW ROW as nrow
FOR EACH ROW
DECLARE assina_serv NUMBER;
BEGIN
        SELECT COUNT(*) INTO assina_serv
        FROM Assinaturas a1 INNER JOIN Assinaturas a2 USING (idContrato)
        WHERE a1.idServ = nrow.idServPac AND a2.idServ = nrow.idServInc;
        IF assina_serv > 0 THEN
             Raise_Application_Error(-20000,'Não pode assinar pacote e
        serviço');
        END IF;
END;
```