

a) junções de 3 ou mais tabelas, com ORDER BY:

SELECT

Ingresso.id_ingresso,
Ingresso.num_assento,
Ingresso.preco,
Ingresso.setor,
Evento.local AS local_evento,
Evento.data AS data_evento,
Torcedor.nome AS nome_torcedor,
Torcedor.pais AS pais_torcedor

FROM

Ingresso

JOIN

Evento ON Ingresso.id_evento = Evento.id_evento

JOIN

Torcedor ON Ingresso.num_passaporte_torcedor =
Torcedor.num_passaporte_torcedor

ORDER BY

Ingresso.preco DESC;

b) junções de 3 ou mais tabelas, com ORDER BY e filtros na cláusula WHERE

SELECT

Ingresso.id_ingresso,
Ingresso.num_assento,
Ingresso.preco,
Ingresso.setor,
Evento.local AS local_evento,
Evento.data AS data_evento,
Torcedor.nome AS nome_torcedor,

```

    Torcedor.pais AS pais_torcedor
FROM
    Ingresso
JOIN
    Evento ON Ingresso.id_evento = Evento.id_evento
JOIN
    Torcedor ON Ingresso.num_passaporte_torcedor =
Torcedor.num_passaporte_torcedor
WHERE
    Ingresso.preco > 100.00
ORDER BY
    Ingresso.preco DESC;

```

c) junção de 3 ou mais tabelas, usando os operadores LIKE e BETWEEN

```

SELECT
    Atleta.nome AS nome_atleta,
    Atleta.nacionalidade,
    Atleta.altura,
    Atleta.peso,
    ComitOlimpicoNacional.nome AS nome_con,
    ComitOlimpicoInternacional.id_coi
FROM
    Atleta
JOIN
    ComitOlimpicoNacional ON Atleta.id_con = ComitOlimpicoNacional.id_con
JOIN
    ComitOlimpicoInternacional ON ComitOlimpicoNacional.id_coi =
ComitOlimpicoInternacional.id_coi

```

WHERE

Atleta.nome LIKE 'Maria%'

ORDER BY

Atleta.nome;

d) junção de 3 ou mais tabelas, usando os operadores IN e IS NULL/IS NOT NULL

SELECT

Evento.id_evento,

Evento.local,

Evento.data,

Estadio.nome AS nome_estadio,

Estadio.local AS local_estadio,

Modalidades.nome AS nome_modalidade,

Modalidades.tipo AS tipo_modalidade

FROM

Evento

LEFT JOIN

Estadio ON Evento.id_estadio = Estadio.id_estadio

JOIN

Modalidades ON Evento.id_modalidade = Modalidades.id_modalidade

WHERE

Evento.local IN ('Rio de Janeiro', 'Brasília')

AND Estadio.id_estadio IS NOT NULL

ORDER BY

Evento.data;

e) junção de 2 ou mais tabelas com GROUP BY, sem HAVING, usando uma função agregada

qualquer (MIN, MAX, AVG, SUM, COUNT). Use ORDER BY

```
SELECT
    Atleta.nacionalidade,
    AVG(Atleta.altura) AS altura_media
FROM
    Atleta
JOIN
    ComitOlimpicoNacional ON Atleta.id_con = ComitOlimpicoNacional.id_con
GROUP BY
    Atleta.nacionalidade
ORDER BY
    altura_media DESC;
```

f) junção de 2 ou mais tabelas com GROUP BY e HAVING, usando uma função agregada qualquer (MIN, MAX, AVG, SUM, COUNT)

```
SELECT
    Modalidades.nome AS nome_modalidade,
    AVG(Atleta.altura) AS altura_media
FROM
    Atleta
JOIN
    Competicao ON Atleta.num_passaporte_atleta =
    Competicao.num_passaporte_atleta
JOIN
    Modalidades ON Competicao.id_modalidade = Modalidades.id_modalidade
GROUP BY
    Modalidades.nome
HAVING
    AVG(Atleta.altura) < 1.8
```

ORDER BY

altura_media DESC;

g) subselect sem correlação

SELECT

nome,

capacidade

FROM

Estadio

WHERE

capacidade > (SELECT AVG(capacidade) FROM Estadio);

h) subselect com correlação

SELECT

a.nome,

a.peso,

a.nacionalidade

FROM

Atleta a

WHERE

a.peso > (SELECT AVG(a2.peso)

FROM Atleta a2

WHERE a2.id_con = a.id_con);

i) subselect com EXISTS

SELECT

cn.id_con,

cn.nome

```
FROM
    ComitOlimpicoNacional cn
WHERE
    EXISTS (SELECT 1
        FROM Atleta a
        WHERE a.id_con = cn.id_con);
```

j) junção de 2 ou mais tabelas com GROUP BY e HAVING, usando uma função agregada qualquer (MIN, MAX, AVG, SUM, COUNT)

```
SELECT
    ComitOlimpicoNacional.nome AS nome_con,
    COUNT(Atleta.num_passaporte_atleta) AS num_atletas
FROM
    Atleta
JOIN
    ComitOlimpicoNacional ON Atleta.id_con = ComitOlimpicoNacional.id_con
GROUP BY
    ComitOlimpicoNacional.nome
HAVING
    COUNT(Atleta.num_passaporte_atleta) > 1
ORDER BY
    num_atletas DESC;
```