

# Banco de Dados

Rebeca Barros



# Trabalhando com Conjuntos

MySQL



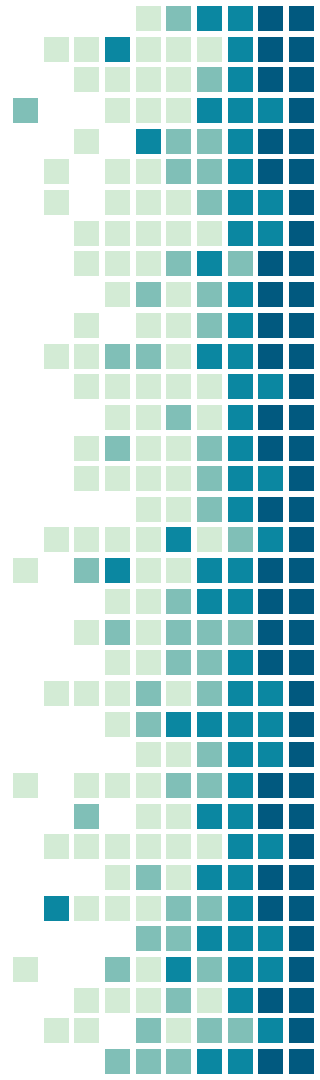
# Conjuntos

- Embora seja possível interagir com os dados uma linha por vez, bancos relacionais dizem respeito a conjuntos de dados. O SQL permite combinar múltiplas tabelas usando vários operadores de conjuntos baseados na teoria matemática.
- São eles: **União, Intersecção e Exceção.**

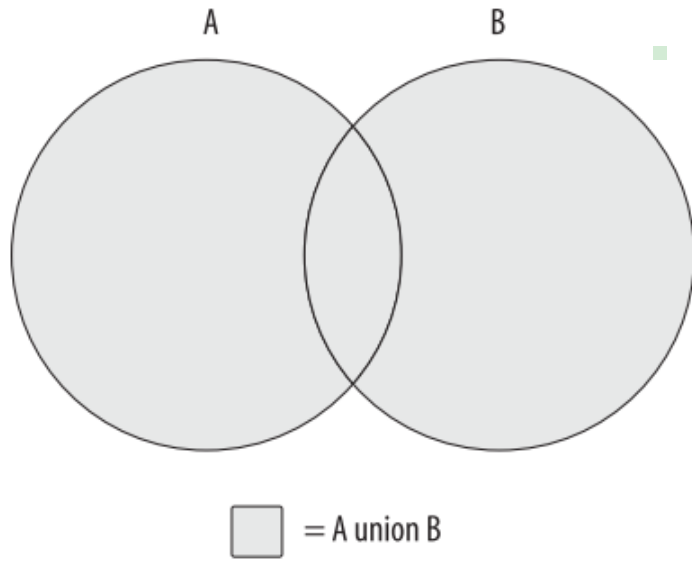


# Conjuntos

- Quando for realizar operações envolvendo dois ou mais conjuntos de dados é preciso seguir as seguintes instruções:
  - Ambos os conjuntos devem possuir o mesmo número de colunas;
  - Os tipos de dados de cada coluna nos dois conjuntos devem ser o mesmo (ou o sgbd deve ser capaz de converter um para o outro).



# União



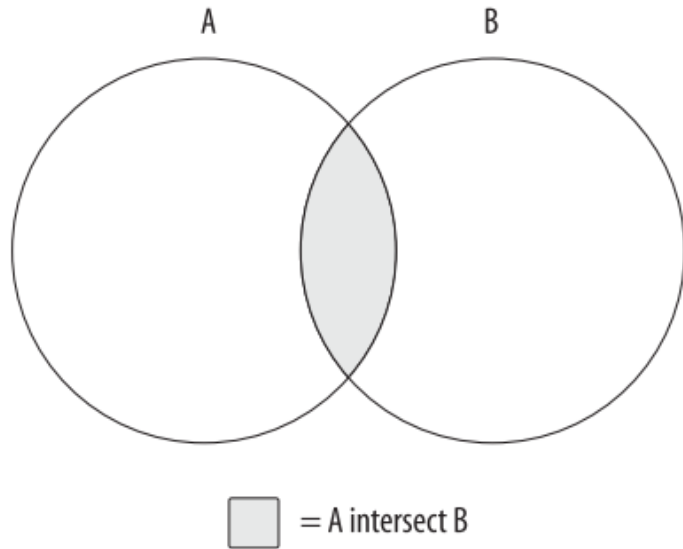
- Os operadores **union** e **union all** permitem combinar múltiplos conjuntos de dados. A diferença entre eles é que o **union** remove os dados duplicados enquanto que o **union all** não.

# União

```
-- exemplo do comando UNION
select idCompositor, nome FROM compositor WHERE idCompositor < 10
UNION
select idCompositor, nome FROM compositor WHERE idCompositor < 15;

-- exemplo do comando UNION ALL
select idCompositor, nome FROM compositor WHERE idCompositor < 10
UNION ALL
select idCompositor, nome FROM compositor WHERE idCompositor < 15;
```

# Intersecção



- A especificação do SQL inclui o operador **intersect** e **intersect all** para realizar operações de intersecção.
- Infelizmente o MySQL não implementa esse operador. SGBDs como Oracle e SQL Server dispõe desse operador para realizar intersecções.

# Intersecção

*-- exemplo do comando INTERSECT - não implementado no MySQL*

```
select Gravadora_idGravadora from cd  
INTERSECT  
select idGravadora from gravadora;
```

*-- exemplo do comando INTERSECT ALL - não implementado no MySQL*

```
select Gravadora_idGravadora from cd  
INTERSECT ALL  
select idGravadora from gravadora;
```

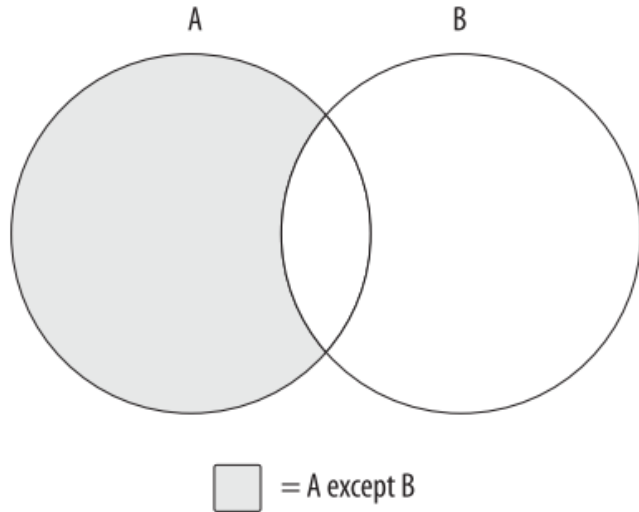


# Intersecção – Simulando no MySQL

```
-- simulando intersecção com inner join e distinct  
select distinct Gravadora_idGravadora FROM cd inner join  
gravadora on cd.Gravadora_idGravadora = gravadora.idGravadora;
```

```
-- simulando intersecção com in e subquery  
select distinct Gravadora_idGravadora FROM cd  
where gravadora_idGravadora in  
(select idGravadora from gravadora);
```

# Exceção/Minus



- A especificação do SQL inclui o operador **except** e **except all** para realizar operações de exceção, ou seja, quando as linhas existem em um conjunto e não existem no outro. Infelizmente o MySQL também não implementa esse operador. Em alguns casos, como no Oracle, o operador para realizar essa operação recebe o nome de *minus*.

# Exceção/Minus

```
-- exemplo do comando EXCEPT - não implementado no MySQL
select idGravadora from gravadora
EXCEPT
select Gravadora_idGravadora from cd;
```

```
-- exemplo do comando EXCEPT ALL - não implementado no MySQL
select idGravadora from gravadora
EXCEPT ALL
select Gravadora_idGravadora from cd;
```

## Exceção – Simulando no MySQL

```
-- simulando EXCEPT com LEFT JOIN  
select idGravadora from gravadora left outer join cd  
ON gravadora.idGravadora = cd.Gravadora_idGravadora  
WHERE cd.Gravadora_idGravadora IS NULL;
```

# Referências

- Beaulieu, A. (2009). *Learning SQL: Master SQL Fundamentals*. "O'Reilly Media, Inc."
- de OLIVEIRA, C. H. P. (2002). *SQL: curso prático*. Novatec.
- Documentação oficial do MySQL. Disponível em: <https://dev.mysql.com/doc/refman/8.0/en/>
- Curso Completo de MySQL (Bóson Treinamentos). Disponível em: <http://www.bosontreinamentos.com.br/curso-completo-de-mysql/>