ANÁLISE GEOESPACIAL EM BANCOS DE DADOS

Parte I – Introdução ao SQL.

Agosto de 2018
Brasília-DF

Luiz.Motta@ibama.gov.br
CENIMAG/IBAMA

Conteúdo

- O que é SQL.
- Divisões do SQL
- Tipo de dados
- Tabela
- Essência da Pesquisa (Query)
- Predicados.
- Exibindo as consultas
- Tipos de funções
- Agrupando dados
- Agrupando tabelas
- Relacionando tabelas
- Subqueries
- Visões x Criar tabela

O que é SQL.

• SQL (Structured Query Language) é uma linguagem padrão para manipular (definir, modificar e consultar) dados.

• Criado pela IBM (1970)

• Existe diferentes versões da linguagem (ANSI)

• Utilizada pelo RDBMS (Gerênciadores de banco de dados)

https://pt.wikipedia.org/wiki/SQL

Divisões do SQL

- DDL (Data Definition Language): **ESTRUTURA**
CREATE, ALTER e DROP

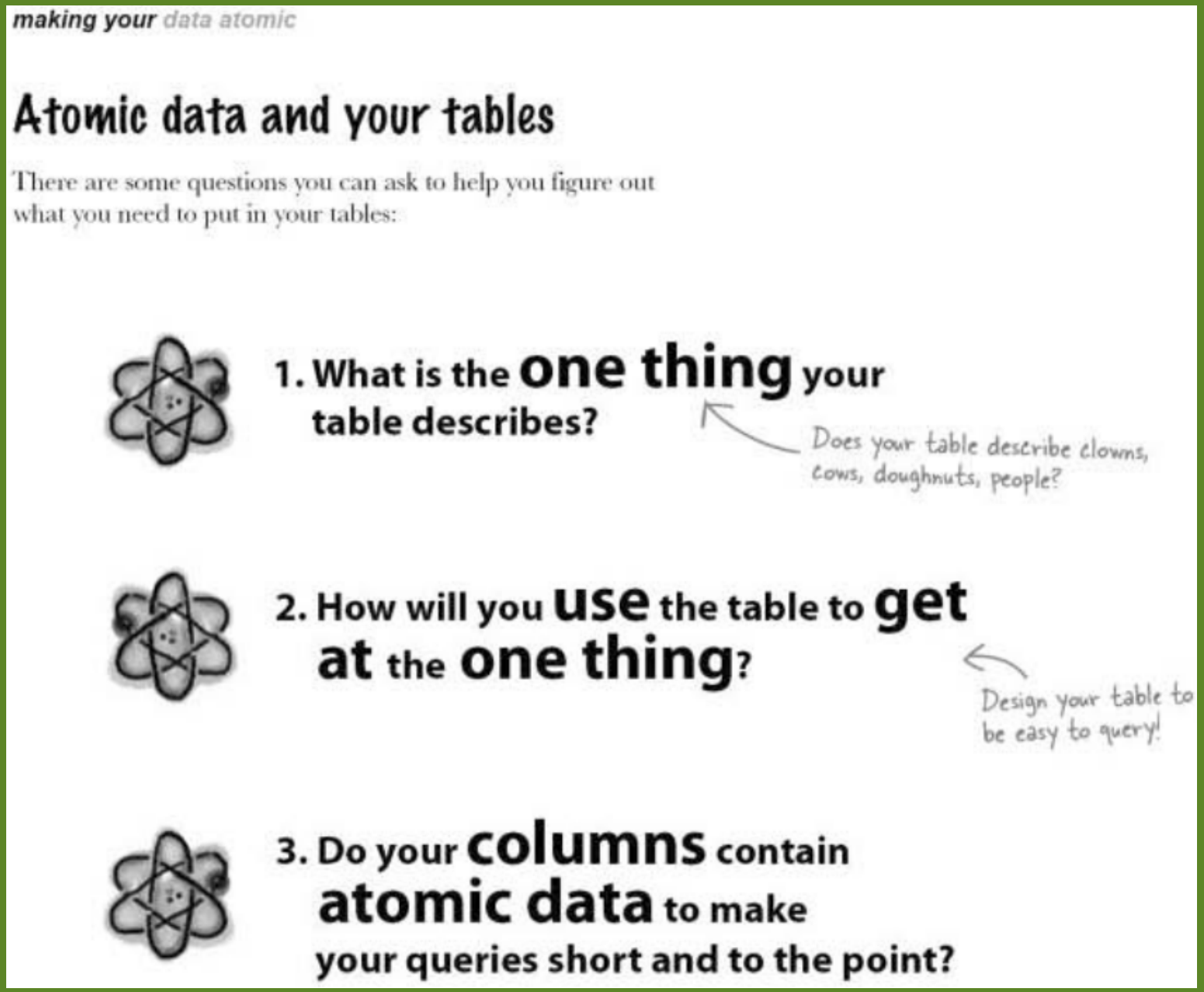
- DML (Data Manipulation Language): **DADOS**
INSERT, UPDATE, DELETE e **SELECT**.

- DCL(Data Control Language): **USUÁRIOS**
GRANT, REVOKE

Tabela

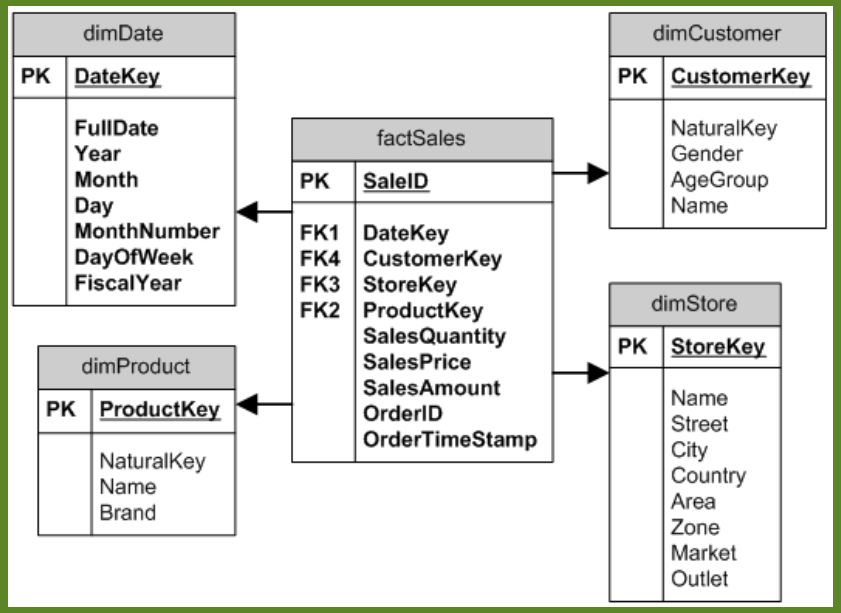
- Onde as informações são organizadas no banco de dados (container de tabelas).
- Organizada na forma de colunas(campos) e linhas (valores/atributos).
- Os campos de uma tabela, definem uma categoria de informações. Os valores são exemplos de categoria.
- Relações entre as tabelas (chaves primária e estrangeira): um-um, um-muitos, muitos-muitos.
- Planejamento do BD.
  - Tabela: Pense para que serve a tabela, a essência dela. Não existe APENAS uma tabela. Separe os dados em categorias. Deve-se definir o mínimo de campos numa tabela(“Atomicidade”).
  - Campo: Descreva literalmente o campo, escolha exemplos(valores), depois, defina o tipo

Tabela(cont.)



O'Reilly - Head First SQL, Lynn Beighley

Tabela(cont)




```sql
CREATE TABLE ibama.img_catalogo_landsat_a
(
  objectid integer NOT NULL DEFAULT nextval('ibama.catalogo_landsat_objectid_seq'::regclass),
  geom geometry(MultiPolygon,4674),
  path character varying(80),
  image character varying(80),
  url_tms character varying(99),
  data date,
  nuvens numeric(12,8),
  quicklook character varying(150),
  orbita character varying(3),
  ponto character varying(3),
  url_tms_interno character varying(99),
  CONSTRAINT catalogo_landsat_pk PRIMARY KEY (objectid),
  CONSTRAINT enforce_dims_shape CHECK (st_ndims(geom) = 2),
  CONSTRAINT enforce_srid_shape CHECK (st_srid(geom) = 4674)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE ibama.img_catalogo_landsat_a
  OWNER TO postgres;
GRANT ALL ON TABLE ibama.img_catalogo_landsat_a TO postgres;
GRANT SELECT ON TABLE ibama.img_catalogo_landsat_a TO indicar_landsat;
GRANT SELECT ON TABLE ibama.img_catalogo_landsat_a TO gestor;
GRANT SELECT ON TABLE ibama.img_catalogo_landsat_a TO ibama;
GRANT ALL ON TABLE ibama.img_catalogo_landsat_a TO indicarprocess;
GRANT SELECT ON TABLE ibama.img_catalogo_landsat_a TO geoserver;
```

create-table.sql

```sql
CREATE TABLE public.t_import_array
(
    "ID" serial NOT NULL,
    char_array character varying(50)[],
    nummeric_array numeric(18, 2)[],
    timestamp_array timestamp with time zone[],
    PRIMARY KEY ("ID")
)
WITH (
    OIDS = FALSE
);

ALTER TABLE public.t_import_array
    OWNER to postgres;
```

```sql
SELECT image, orbita, ponto, data
FROM ibama.img_catalogo_landsat_a
LIMIT 5
```

ut pane

ta Output | Explain | Messages | History

	image character varying(80)	orbita character varying(3)	ponto character varying(3)	data date
1	LE70010572013005CUB00_r5g4b3.tif	001	057	2013-01-05
2	LC82230782014131LGN00_r6g5b4.tif	223	078	2014-05-11
3	LC82230772015214LGN00_r6g5b4.tif	223	077	2015-08-02
4	LC82230762014259LGN00_r6g5b4.tif	223	076	2014-09-16
5	LC82230772013160LGN00_r6g5b4.tif	223	077	2013-06-09

http://postgresguide.com/tips/dates.html

Essência da Pesquisa

SELECT (expressão)
FROM (tabelas, visões, query)
WHERE (Predicado)

Exemplos:

```
Select 12 – 7;

Select Now() + Interval '7' DAY;

Select 'Imagem = ' || image || ' e a data = ' || data
From ibama.img_catalogo_landsat_a
WHERE
   orbita = '217' AND
   data BETWEEN date '2017-01-01' AND date '2017-02-01';
```

http://www.postgresqltutorial.com/postgresql-between/

Predicados

• **Expressão**: é um item (coluna, constante, retorno de função, ...), ou uma combinação de itens e operadores, que gera um valor único.

Ex.:
```
Select nuvens, nuvens ^3
From ibama.img_catalogo_landsat_a
Limit 2;
```

• NULL ? É a ausência de valor de um campo.

O valor NÃO é conhecido (=, <>, >, <, ..).

Na criação da tabela pode-se definir que o preenchimento é obrigatório, e ainda, com valor padrão.

Ex.: peso DEC(4,2) NOT NULL DEFAULT 00.00

• **Predicado**: é uma expressão avaliada como V, F ou desconhecido.

Usado nas cláusulas WHERE e HAVING.

Predicados (cont.)

Tipos de predicados:

• Relacionais: expressão <operador> expressão

operadores: =, <>, >, <, …

Select 2 > 3; Ex.: qgis_intersects_geojson.sql

• Between: *Select* 8 Between 8 And 10 -- 8 <= AND 8 <= 10

• NULL:

Select *
From ibama.img_catalogo_landsat_a
Where geom Is NULL;

• LIKE: expressão [Not] LIKE 'formato de pesquisa'.

'%' zero ou mais caracteres, '_' Apenas UM.

Select 'ola' like 'ol_'

• SIMILAR: expressão SIMILAR TO 'expressão regular'

Select 'QGIS 3' similar to ('QGIS (1|2|3)')

• In: expressão [Not] In <subquery> ou <lista>

Select 5 In (2, 3, 4, 6) -- 5 = 2 OR 5 = 3 OR ...

Predicados (cont.)

- Precedência : Ordem dos operadores.

Usar parênteses para alterar a ordem.

Ex.: 2+3*4 = 20 → ??

- Conectores de predicados AND e OR:

Predicados são avaliados da esquerda p/ direita

  - AND: Retorna Falso na **primeira** ocorrência.
  - OR: Retorna Falso se **todas** ocorrências.
  - Usar os predicados mais rápidos no início.

    Exemplo: alerta_landsat.sql

- Campo [Operador] Valor
  - Operador compatível com Campo x Valor
  - Errado x Incoerente (Ex.: predicado_incoerente.sql)
- Texto x Campo
  - Aspas simples (' ') delimita um Valor de um texto.
  - Dupla aspas (“ ”) delimita um Objeto(Campo, Esquema, ...)

    Ex.: FROM "cb"."lim_municipio_a"

Predicados (cont.)

Expressões condicionais – Case

- Padrão

Case When condição Then resultado
[When …]
[Else resultado]
End

```
SELECT
 CASE
   WHEN 1 = 1 THEN '1=1'
   WHEN 2 = 2 THEN '2=2'
   ELSE 'SEM'
 END
```

- Forma simplificada para valores

Case expressão
When valor Then resultado
[When …]
[Else resultado]
End

Exemplo: case.sql

Predicados (cont.)

Expressões condicionais

• Coalesce: Retorna o primeiro valor não nulo de seus argumentos.

```
SELECT Coalesce(NULL, 'Sem Valor');
```

• Nullif: Retorna Null se os DOIS valores foram iguais, caso contrário, retorna o PRIMEIRO.

```
SELECT NullIf(1,2)  IS NULL
SELECT COUNT(1)
FROM public.veg_terraclass_cerrado_a
WHERE NULLIF( classe, 'SILVICULTURA') IS NULL AND classe IS NOT NULL
```

• Greatest e Least: Seleciona o maior ou menor valor de uma lista.

```
SELECT Greatest (1, 2, 3, 4, 5);
```

Exibindo consultas

Uso de álias (AS).

- Colunas: Nome p/ coluna numa tabela e/ou apelido.
- Tabelas: apelido

Ex.: SELECT t.campo AS “Descrição” FROM tabela AS t

- Ordenamento de campos
  - Order By Campo1, Campo2, … [ASC / DESC]

```
SELECT campo1, campo2
FROM tabela
ORDER BY campo1, campo2 desc;
```

- Variáveis especiais:
  - SESSION_USER, CURRENT_TIMESTAMP, CURRENT_DATE
- Limitando as linhas: Limit

Tipos de funções

Função: É um tipo de objeto do BD que recebe argumentos(parâmetros) e retornam valores. Ex.: bit_length('12345')

- Funções ANSI SQL e específicas do RDBMS.

Exemplos: matemáticas, string, date/time, …, GEO

- Funções agregadas: O argumento é um conjunto de dados
  - Count:

    ```
    SELECT Count(*), Count(dominio_as)
    FROM ibama.alerta_deter_2017_a;
    ```

  - Sum, Avg, Max, Min, Stddev:

    ```
    SELECT objectid, data_imagem, area_km2,geom
    FROM ibama.alerta_deter_2017_a
    WHERE area_km2 = ( SELECT Max(area_km2) FROM
    ibama.alerta_deter_2017_a );
    ```

- Uso de CAST:

```
SELECT '25.38'::real as numero, Cast('25.38' As Real) as numero;
SELECT 'Valor = ' || 25.38 AS Obs;
```

Tipos de funções(cont.)

Função para String

- Select Char_length('12345');
- Select Lower('Area em Hectares');
- Select Upper('Area em Hectares');
- Select Overlay('Q123456S' placing 'GI' From 2 for 6);
- Select Position('is' in 'QGIS');
- Select Substring('BR 319' from 4 for 3);
- Select Trim(' BR 319 '),
  Trim(Leading ' BR 319 '), Trim(Trailing ' BR 319 ');
- Select Btrim('*&$BR&319&$', '$&*');
- Select Convert('text_in_utf8', 'UTF8', 'LATIN1');
- Select Pg_client_encoding()
- Select Initcap('qgis developer meeting')

http://www.postgresql.org/docs/9.1/static/functions-string.html

Tipos de funções(cont.)

Função para Números

- Select Abs(-58), Power(2,8), Mod(5, 2), Sqrt(16);
- Select pi()--, Floor(pi()), Floor(-1*pi());

http://www.postgresql.org/docs/9.1/static/functions-math.html.

Funções para Datas/Tempo

- Select Current_date, Current_time, Current_timestamp
- Select Localtime, Localtimestamp;
- Select Date '2001-09-28', Interval '1 day';
- Select Timestamp '2001-09-28 01:00', Time '01:00';
- Select Age(Timestamp '2001-04-10', Timestamp '1957-06-13');
- Select Age(Timestamp '2001-04-10');
- Select Clock_timestamp();
- Select Date_part('hour', Timestamp '2001-02-16 20:38:40');
- Select Date_part('month', Interval '2 years 3 months');

Tipos de funções(cont.)

Funções para Datas/Tempo (cont.)

- Select Justify_days(interval '35 days'),
  Justify_hours(interval '27 hours'),
  Justify_interval(interval '1 mon -1 hour');
- Select Statement_timestamp(), Timeofday();

Operadores para Data/Tempo

- Select date '2001-09-28' + integer '7';
- Select date '2001-09-28' + interval '1 hour';
- Select date '2001-09-28' + time '03:00';
- Select date '2001-10-01' - date '2001-09-28';
- Select date '2001-09-28' - interval '1 hour';
- Select 21 * interval '1 day';
- Select (Date '2001-02-16', Date '2001-12-21')
  Overlaps
  (Date '2001-10-30', Date '2002-10-30');
- Select Extract(Day From Timestamp '2001-09-28 01:00');

http://www.postgresql.org/docs/9.1/static/functions-datetime.html

Agrupando dados

• Distinct: Lista os valores distintos de campo(s).

```
SELECT DISTINCT classes
FROM public.veg_cobertura_uso_da_terra_a
ORDER BY classes

SELECT COUNT(DISTINCT classes)
FROM public.veg_cobertura_uso_da_terra_a
```

• Group By: Agrupa registros com mesmos valores segundo expressões agrupadas.

“Sumarizar” -> Uso de funções de agregação

Uso do Having p/ filtrar resultados da agregação.

```
SELECT classes, COUNT(1) AS Total
FROM public.veg_cobertura_uso_da_terra_a
GROUP BY classes
HAVING COUNT(1) < 10000
ORDER BY classes  --   DESC
```

Agrupando tabelas

As tabelas (resultado das seleções) devem ter os mesmos tipos de campos, ou compatíveis(Cast), e estarem na mesma ordem.

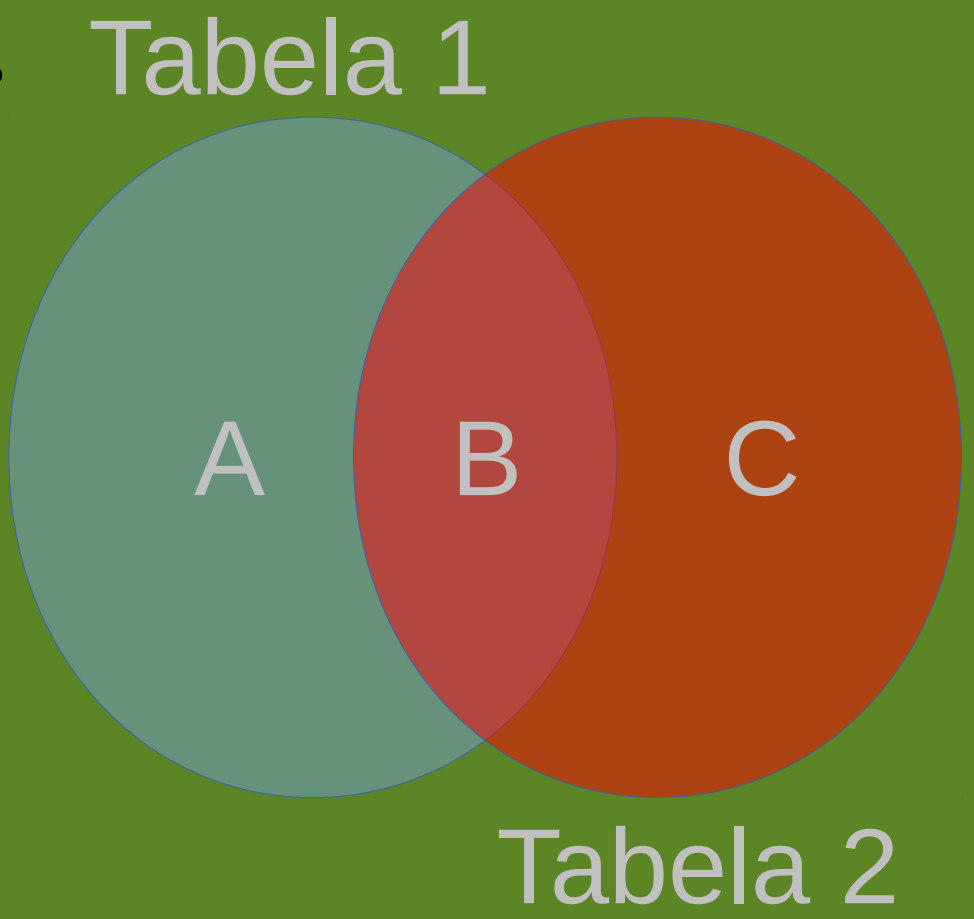
- Union: Tab1 Union Tab2 = A + B + C

Não possuem linhas repetidas.
Usar Union All para repetir as linhas

- Intersect: Tab1 Intersect Tab2 = B

- Except: Tab1 Except Tab2 = A

Exemplo: agreg-tab.sql



Relacionando tabelas

- O relacionamento (join) “combina” valores de mais de uma tabela.
- Utiliza-se predicados para filtrar os valores combinados

Where t1.id = t2.id
[Tipo de Join] On t1.id = t2.id

- Tipos de Joins:
  - Cross Join: Todas as combinações. “Produto Cartesiano”
    Total = N1 x N2 – “Não usa filtro(Where/Join)”
  - Natural Join: Combinações com os mesmos campos (nome e tipo)
    Total = N1 ∩ N2 – Filtra pelos campos(máximo em comum)
  - Column-name Join: Combinações com campos definidos.
    Total = N1 ∩ N2 – Filtra por campos definidos: Using(nome)

Ex.: join_tipos.sql

Relacionando tabelas (cont.)

- Tipos de Joins com cláusula ON:
  - Inner Join: Combinações que atendem o predicado do ON
    Total = N1 ∩ N2
    join_inner.sql (ver case.sql)
  - Outer Join: As combinações que não atendem o predicado do ON recebem o valor NULL
    join_outer.sql
    - Left (Linhas da tabela da esquerda)
      Total = N1, apenas p/ relação 1 → 1
    - Right (Linhas da tabela da direita)
      Total = N2, apenas p/ relação 1 → 1
    - Full (Linhas da tabela da esquerda e direita)
      Total = N1 + N2 - (N1 ∩ N2), apenas p/ relação 1 → 1
      * Clausula ON apenas com campos relacionáveis

Subqueries

- In: expressão IN (select expressão from …)

SELECT * FROM dominios.revestimento
WHERE code IN (SELECT code FROM dominios.instituicao)

- Expressão [operador] [All/Any/Some] (select expressão from …)
  - All: Predicado Verdadeiro p/ TODOS os regitros.
    SELECT 6 > ALL (SELECT code FROM dominios.revestimento) → True
  - Any ou Some: Predicado Verdadeiro se pelo menos UM registro.
    SELECT 3 = ANY (SELECT code FROM dominios.revestimento) → True

https://www.w3schools.com/sql/sql_any_all.asp

- Exists (select expressão from …)

Predicado Verdadeiro se retornar algum valor.

SELECT EXISTS (SELECT code FROM dominios.revestimento WHERE code = 3)

- UNIQUE (select expressão from …)

Verdadeiro se os registros possuem valores únicos

- Queries aninhadas:
  - Select t1.* From (Select * From t2)t1
  - Select * From t Where c = (Select c From t2 Where predicado)
    *Subquery retorna UM Elemento

Visões x Criar tabela

- View: Registrar uma consulta
  - Pode-se usar como uma tabela (From)
  - Não gera valores.
  - É executada quando utilizada em outra consulta.

Create View esquema.visao AS [Select …];

- Uma tabela pode ser gerada por um 'Select'

Create Table esquema.tabela As [Select];

Tipo de dados

- Booleano: Falso ou Verdadeiro
- Números: Inteiro e Ponto flutuante
- String: Texto
- Data/tempo
- Intervalo
- Geométrico (ponto, linha, polígono, ...)

* Postgis cria o campo GEOM (possui SR)

- Outros: Binário, XML, UUID, Arrays, tipo do usuário,...