

Introdução a Bancos de Dados SQL

Parte 1 – Introdução e SELECT

Clodoveu Davis

Departamento de Ciência da Computação

Universidade Federal de Minas Gerais

clodoveu@dcc.ufmg.br



SQL

- Structured Query Language
 - Desenvolvida nos anos 1970 pela IBM, para o System R
 - Nome original: SEQUEL – Structured English Query Language
 - Até hoje se pronuncia “SEQUEL” em inglês para se referir a SQL
- Foi sendo progressivamente padronizada e expandida
 - SQL 1 (1986) – ANSI/ISSO
 - SQL 2 (1992), virou SQL-92
 - SQL:1999, SQL:2003, ... SQL:2016 (atual)
- Ainda existem extensões e detalhes específicos de cada implementação de SGBD

SQL

- Baseada na álgebra relacional e no cálculo relacional de tuplas
- É uma linguagem declarativa, em que se especifica o resultado desejado, não como obtê-lo
- O processo para obter o resultado é definido pelo processador de consultas do SGBD, de forma transparente
- Opera sobre *tabelas*, formadas por *linhas* e *colunas*

ER	Conceitual	Entidade	Instância	Atributo
Relacional	Lógico	Relação	Tupla	Atributo
SQL	Físico	Tabela	Linha	Coluna

SQL

DML

Data Management Language

INSERT
DELETE
UPDATE

DDL

Data Definition Language

CREATE TABLE
CREATE INDEX
CREATE VIEW
ALTER TABLE
ALTER INDEX
DROP TABLE
DROP INDEX
DROP VIEW

DCL

Data Control Language

GRANT
REVOKE

DTL

Data Transaction Language

BEGIN TRANSACTION
COMMIT
ROLLBACK

DQL

Data Query Language

SELECT

Exemplo

FILME		
ID	Titulo	AnoProd
001	Amadeus	1984
002	Birdman	2014
003	Casino	1995
004	Die Hard	1988
005	El Cid	1961
006	Fargo	1996
007	Grease	1978
008	Hair	1979
009	It	2017
010	JFK	1991
011	Kundun	1997
012	Lincoln	2012
013	Misery	1990
014	Nixon	1995
015	Outbreak	1995
016	Psycho	1960
017	Quiz Show	1994
018	Rush	2013
019	Shrek	2001
020	Twister	1996
021	Up	2009
022	Vertigo	1958
023	WarGames	1983
024	X-Men	2000
025	Yesterday	2019
026	Zootopia	2016
027	The Godfather	1972
028	The Shawshank Redemption	1994
029	Goodfellas	1990
030	The Godfather Part II	1974

PESSOA		
IdP	Nome	AnoNasc
111	Charlton Heston	1923
114	Dustin Hoffman	1937
115	Lily James	1989
117	Frances McDormand	1957
118	Alfred Hitchcock	1899
151	Shakira	1977
207	Kathy Bates	1948
208	Sharon Stone	1958
215	Hugh Jackman	1968
217	Anthony Perkins	1932
221	Francis Ford Coppola	1939
222	Robert DeNiro	1943
229	James Caan	1940
245	Al Pacino	1940
260	Ralph Fiennes	1962
276	Stephen King	1947
279	Tom Hulce	1953
281	Oliver Stone	1946
298	Morgan Freeman	1937
311	Helen Hunt	1963
324	Daniel Day-Lewis	1957
338	Milos Forman	1932
342	Martin Scorsese	1942
352	Mike Myers	1963
356	Bruce Willis	1955
367	Matthew Broderick	1962
368	Famke Janssen	1964
382	Emma Stone	1988
427	John Travolta	1954
428	Tim Robbins	1958
438	Kevin Costner	1955
439	Chris Hemsworth	1983
446	Anthony Hopkins	1937
453	Edward Asner	1929
456	Olivia Newton-John	1948
495	Michael Keaton	1951
900	Joel Coen	1954
901	F. Murray Abraham	1939
907	Marlon Brando	1924
950	Robert Redford	1936
999	Steven Spielberg	1946

FP		
ID	N	Activity
001	338	Director
001	279	Actor
001	901	Actor
002	495	Actor
002	382	Actor
003	208	Actor
003	222	Actor
004	356	Actor
005	111	Actor
006	117	Actor
006	900	Director
007	427	Actor
007	456	Actor
008	338	Director
009	276	Writer
010	438	Actor
010	281	Director
011	342	Director
012	324	Actor
012	999	Director
013	207	Actor
013	229	Actor
013	276	Writer
014	446	Actor
015	114	Actor
016	118	Director
016	217	Actor
017	260	Actor
017	950	Director
018	439	Actor
019	352	Actor
020	311	Actor
021	453	Actor
022	118	Director
023	367	Actor
024	215	Actor
024	368	Actor
025	115	Actor
026	151	Actor
027	222	Actor
027	229	Actor
027	245	Actor
027	221	Director
027	907	Actor
028	428	Actor
028	298	Actor
029	222	Actor
029	342	Director
030	245	Actor
030	222	Actor
030	221	Director
030	907	Actor

PREMIACAO			
ID	Sigla	IdP	AnoPremio
001	OBP	NULL	1984
001	OBA	901	1984
002	OBP	NULL	2014
002	NBA	495	1984
003	NBA	208	1995
006	NBP	NULL	1996
006	NBD	900	1996
010	NBP	NULL	1991
010	NBD	281	1991
012	NBP	NULL	2012
012	OBA	324	2012
012	NBD	999	2012
013	OBA	207	1990
016	NBD	118	1960
017	NBP	NULL	1994
017	NBD	950	1994
021	NBP	NULL	2009
027	OBP	NULL	1972
027	OBA	907	1972
027	NBD	221	1972
028	NBP	NULL	1994
030	OBP	NULL	1974
030	NBA	245	1974
030	OBD	221	1974

PREMIO	
Sigla	NomePremio
NBA	Oscar Nominee Best Actor/Actress
NBD	Oscar Nominee Best Director
NBP	Oscar Nominee Best Picture
OBA	Oscar Best Actor
OBD	Oscar Best Director
OBP	Oscar Best Picture

DQL: comando SELECT

SELECT <lista de atributos>

FROM <lista de tabelas>

WHERE <condição>

Onde <condição> é um predicado booleano

SELECT e a álgebra relacional

SELECT *
FROM <tabela>
WHERE <condição>

(* = todos os atributos)

$\sigma_{\text{condição}}(\text{tabela})$

SELECT <lista atributos>
FROM <tabela>

$\pi_{\text{lista atributos}}(\text{tabela})$


```
SELECT titulo, anoprod
FROM filme
WHERE anoprod > 2000
```

	titulo character varying (100)	anoprod integer
1	Birdman	2014
2	It	2017
3	Lincoln	2012
4	Rush	2013
5	Shrek	2001
6	Up	2009
7	Yesterday	2019
8	Zootopia	2016

```
SELECT *
FROM filme
```

	id [PK] integer	titulo character varying (100)	anoprod integer
1	1	Amadeus	1984
2	2	Birdman	2014
3	3	Casino	1995
4	4	Die Hard	1988
5	5	El Cid	1961
6	6	Fargo	1996
7	7	Grease	1978
8	8	Hair	1979
9	9	It	2017
10	10	JFK	1991
11	11	Kundun	1997
12	12	Lincoln	2012
13	13	Misery	1990
14	14	Nixon	1995
15	15	Outbreak	1995

SELECT e a álgebra relacional

SELECT <lista atributos>
FROM <tabela>
WHERE <condição>

$\pi_{\text{lista atributos}} \sigma_{\text{condição}}(\text{tabela})$

SELECT <lista atrib>
FROM <tab1, tab2>
WHERE <condição>

$\pi_{\text{lista atrib}} \sigma_{\text{condF}}(\text{tab1} \bowtie_{\text{condJ}} \text{tab2})$

<condição> = <condJunção> AND <condFiltro>

```
SELECT filme.titulo, pessoa.nome
FROM filme, fp, pessoa
WHERE filme.id = fp.id
AND fp.idp = pessoa.idp
```

Três tabelas

Cláusula de junção
com duas condições

	titulo character varying (100)	nome character varying (100)
1	Amadeus	F. Murray Abraham
2	Amadeus	Milos Forman
3	Amadeus	Tom Hulce
4	Birdman	Michael Keaton
5	Birdman	Emma Stone
6	Casino	Robert DeNiro
7	Casino	Sharon Stone
8	Die Hard	Bruce Willis
9	El Cid	Charlton Heston
10	Fargo	Joel Coen
11	Fargo	Frances McDormand
12	Grease	Olivia Newton-John
13	Grease	John Travolta
14	Hair	Milos Forman
15	It	Stephen King

Mas por que está ordenado se não usei ORDER BY?

Não há garantia que o resultado será ordenado. Isso dependerá da ordenação das linhas no disco, do algoritmo de processamento de consultas usado pelo SGBD, e outros fatores.

```
SELECT filme.titulo, pessoa.nome  
FROM filme, fp, pessoa  
WHERE filme.id = fp.id  
AND fp.idp = pessoa.idp  
AND filme.anoprod = 1995
```

Cláusula de filtro

	titulo character varying (100) 🔒	nome character varying (100) 🔒
1	Casino	Sharon Stone
2	Casino	Robert DeNiro
3	Nixon	Anthony Hopkins
4	Outbreak	Dustin Hoffman

SELECT e a álgebra relacional

SELECT <lista atributos>

FROM <tabela>

WHERE <condição>

$\pi_{\text{lista atributos}} \sigma_{\text{condição}}(\text{tabela})$

SELECT <lista atrib>

FROM <tab1, tab2>

$\pi_{\text{lista atrib}} (\text{tab1 X tab2})$

(cláusula WHERE ausente == TRUE)

```
SELECT filme.titulo, pessoa.nome
FROM filme, fp, pessoa
WHERE filme.anoprod = 1995
```

Sem cláusula de junção:
produto cartesiano

	titulo character varying (100)	nome character varying (100)
1	Casino	Charlton Heston
2	Nixon	Charlton Heston
3	Outbreak	Charlton Heston
4	Casino	Charlton Heston
5	Nixon	Charlton Heston
6	Outbreak	Charlton Heston
7	Casino	Charlton Heston
8	Nixon	Charlton Heston
9	Outbreak	Charlton Heston
10	Casino	Charlton Heston
11	Nixon	Charlton Heston
12	Outbreak	Charlton Heston
13	Casino	Charlton Heston
14	Nixon	Charlton Heston

SELECT

- Nomes de atributos ambíguos
 - Prefixar com o nome da tabela: TABELA.atributo
- Mudança de nomes (álgebra relacional: RENAME)
 - Cláusula AS

```
SELECT E.nome, E.salario  
FROM EMPREGADO AS E
```

- AS pode ser usada para renomear todos os atributos também

```
FROM EMPREGADO AS E(n, sal, dn, end)
```

- AS pode ser omitida

```
FROM EMPREGADO E
```

```
SELECT filme.titulo, pessoa.nome  
FROM filme, fp, pessoa  
WHERE filme.id = fp.id  
AND fp.idp = pessoa.idp  
AND filme.anoprod = 1995
```

```
SELECT f.titulo, p.nome  
FROM filme f, fp x, pessoa p  
WHERE f.id = x.id  
AND x.idp = p.idp  
AND f.anoprod = 1995
```

```
SELECT titulo, nome  
FROM filme f, fp x, pessoa p  
WHERE f.id = x.id  
AND x.idp = p.idp  
AND anoprod = 1995
```




Ordenação de resultados

- Ordenação
 - No modelo relacional, não há ordenação (lógica de conjuntos)
- Cláusula ORDER BY <expressão> [ASC | DESC]
 - ASC é default

```
SELECT *  
FROM EMPREGADO  
WHERE dept = 5  
ORDER BY salario DESC
```

Critérios múltiplos: ORDER BY salario DESC, nome ASC

```
SELECT titulo, nome
FROM filme f, fp x, pessoa p
WHERE f.id = x.id
AND x.idp = p.idp
AND anoprod = 1995
ORDER BY nome
```

	titulo character varying (100) 	nome character varying (100) 
1	Nixon	Anthony Hopkins
2	Outbreak	Dustin Hoffman
3	Casino	Robert DeNiro
4	Casino	Sharon Stone

SQL

- As relações do modelo relacional se comportam como conjuntos e não admitem tuplas repetidas
- Em SQL, as tabelas funcionam como **multiconjuntos**, i.e., linhas duplicadas podem existir e podem estar no resultado de uma consulta
 - Eliminar linhas duplicadas tem custo computacional
 - Pode ser de interesse do usuário ver as duplicatas
 - As duplicatas podem posteriormente entrar no cálculo de operações de agregação (SUM, COUNT, etc.)
- Eliminar duplicatas: `SELECT DISTINCT`

```

SELECT nome
FROM filme f, fp x, pessoa p
WHERE f.id = x.id
AND x.idp = p.idp
AND anoprod BETWEEN 1990 AND 1999
ORDER BY nome

```

	nome character varying (100)
1	Anthony Hopkins
2	Dustin Hoffman
3	Frances McDormand
4	Helen Hunt
5	James Caan
6	Joel Coen
7	Kathy Bates
8	Kevin Costner
9	Martin Scorsese
10	Martin Scorsese
11	Morgan Freeman
12	Oliver Stone
13	Ralph Fiennes
14	Robert DeNiro
15	Robert DeNiro

```

SELECT DISTINCT nome
FROM filme f, fp x, pessoa p
WHERE f.id = x.id
AND x.idp = p.idp
AND anoprod BETWEEN 1990 AND 1999
ORDER BY nome

```

	nome character varying (100)
1	Anthony Hopkins
2	Dustin Hoffman
3	Frances McDormand
4	Helen Hunt
5	James Caan
6	Joel Coen
7	Kathy Bates
8	Kevin Costner
9	Martin Scorsese
10	Morgan Freeman
11	Oliver Stone
12	Ralph Fiennes
13	Robert DeNiro
14	Robert Redford

Operações de conjuntos

- Comandos UNION, INTERSECT, EXCEPT podem ser aplicados entre tabelas (ou resultados de consultas) que sejam compatíveis para união
 - No resultado as linhas duplicadas são eliminadas, o que pode afetar o desempenho
 - Usar UNION ALL, INTERSECT ALL, EXCEPT ALL para deixar as duplicatas

```
SELECT cidade  
FROM professor p  
  
UNION
```

```
SELECT município  
FROM aluno a      --- produz uma lista de cidades distintas  
                   onde moram professores ou alunos
```

```
select nome
from filme f, pessoa p, fp x
where f.id = x.id
and p.idp = x.idp
and anoprod - anonasc > 50
```

INTERSECT

```
select nome
from premio a, premiacao b, pessoa
where a.sigla = b.sigla
and b.idp = c.idp
```

Pessoas que ganharam prêmios por filmes que fizeram depois dos 50 anos de idade

	nome character varying (100)
1	Milos Forman
2	Michael Keaton
3	Robert DeNiro
4	Stephen King
5	Martin Scorsese
6	Steven Spielberg
7	Daniel Day-Lewis
8	Anthony Hopkins
9	Dustin Hoffman
10	Alfred Hitchcock
11	Robert Redford
12	Edward Asner
13	Alfred Hitchcock
14	Morgan Freeman



	nome character varying (100)
1	F. Murray Abraham
2	Michael Keaton
3	Sharon Stone
4	Joel Coen
5	Oliver Stone
6	Daniel Day-Lewis
7	Steven Spielberg
8	Kathy Bates
9	Alfred Hitchcock
10	Robert Redford
11	Marlon Brando
12	Francis Ford Coppola
13	Al Pacino
14	Francis Ford Coppola

	nome character varying (100)
1	Robert Redford
2	Steven Spielberg
3	Alfred Hitchcock
4	Daniel Day-Lewis
5	Michael Keaton

```
SELECT titulo
FROM filme f, fp x, pessoa p
WHERE f.id = x.id
AND p.idp = x.idp
AND nome LIKE 'R%'
```

INTERSECT

```
SELECT titulo
FROM filme f, premiacao x, premio p
WHERE f.id = x.id
AND p.sigla = x.sigla
AND nomepremio LIKE '%Oscar%'
```

	titulo	
	character varying (100)	🔒
1	Casino	
2	Quiz Show	
3	Quiz Show	
4	The Godfather	
5	Goodfellas	
6	The Godfather Part II	



	titulo	
	character varying (100)	🔒
1	Amadeus	
2	Amadeus	
3	Birdman	
4	Birdman	
5	Casino	
6	Fargo	
7	Fargo	
8	JFK	
9	JFK	

	titulo	
	character varying (100)	🔒
1	The Godfather	
2	The Godfather Part II	
3	Quiz Show	
4	Casino	

Expressões e operadores

- Operações matemáticas: + - * /
- Concatenação de strings: ||
- Operadores lógicos: AND, OR, NOT
- Comparação: > < >= <= = <> e BETWEEN (ex. WHERE atr **BETWEEN** 1 and 3)
 - Comparação múltipla: atrib **IN** (val1, val2, val3)
 - IS NULL, IS NOT NULL (= e <> não podem ser usados com nulos)
- LIKE: comparação parcial de strings; usar % para vários caracteres e _ para um caractere

```
SELECT cidade, estado  
FROM aluno  
WHERE cidade LIKE 'São%'
```


Para praticar

- GeoSQL: pode ser usado para fazer consultas não-geo
 - <http://aqui.io/geosql>
- SQL Fiddle: você entra com tabelas e dados, e executa operações
 - <http://sqlfiddle.com>
- w3schools.com/sql: tutorial de muitas funções, com exemplos interativos
 - <http://w3schools.com/sql>
- No seu ambiente: instale MySQL ou PostgreSQL, por exemplo.
 - Um dataset está disponível na página da disciplina

