

6. SQL // DML

full qualified name (fqdn)

<tablename>.<attrname>

usado quando há mais de uma tabela na operação com colunas sob o mesmo nome

definição | Data Manipulation Language

- inserir, eliminar e atualizar dados
- efetuar consultas

questões básicas

inserção | inserir novo tuplo numa relação

INSERT INTO <tablename> VALUES (v1, v2, ..., vn);

valores têm de respeitar ordem de criação dos atributos

INSERT INTO <tablename> (A1, A4) VALUES (v1, v4);

cria tuplo com determinados valores (constant NULL or DEFAULT)

obs | podem ser utilizados os termos NULL or DEFAULT
na segunda sintaxe, valores constantes têm de ser definidos!

eliminação | DELETE FROM <tablename> WHERE <attr> = /> / ... <value>;

remoção de tuplo(s) que verifica condição

DELETE FROM <tablename>;

remove todas as tuplos da relação

obs | devido às restrições de integridade referencial (CASCADE)
pode propagar-se a outras relações

atualização | UPDATE <tablename>

SET <attr> = <value>, ...

WHERE <condition>;

atualiza valores de tuplo(s)

obs | tal como na eliminação, pode propagar-se (CASCADE)

consultas simples

projeção | SELECT <attrs> FROM <tablename>;

seleciona um conjunto de atributos de uma tabela

obs | attrs = * seleciona todas as colunas

SELECT ALL seleciona todos os tuplos

SELECT DISTINCT elimina tuplos duplicados

selecção

o

SELECT ... FROM ... WHERE <condition(s)>

permite seleccionar subconjunto de tuplos de acordo com condições

obs

operadores = < > <= >= <> AND OR NOT

renomeação

P

SELECT attr1 AS attr2 FROM table1 AS table2
de atributos de relações

obs

podem "renomear-se" operações sobre atributos

exemplo SELECT salary * 0.35 AS salaryTaxes FROM Employee
no SQL Server, se não renomear mas não é dado nome à coluna ⚠

união.

intersecção
e diferença

U ∩ -

<R1> UNION / INTERSECT / EXCEPT <R2>

as duas relações devem ter o mesmo número de atributos com domínios compatíveis

duplicados

são eliminados por defeito

para evitar problemas usar os operadores

UNION ALL INTERSECT ALL EXCEPT ALL

∩ disponíveis em SQL Server

operação vertical

operação horizontal

produto

cartesiano X

SELECT * FROM <table1>, <table2>, ...

+ eficiente

<table> INNER JOIN <table> ON <condition>

junção

X

SELECT <attrlist> FROM <tablelist> WHERE <joincondition>;

pode ser feita sobre 2 ou + relações, desde que as condições cubram todas as envolvidas

queries

comparação de
strings

LIKE 'pattern'

%. 0 ou + caracteres

_ 1 qualquer carácter

[] 1 dos caracteres dados

[^] nenhum dos car. dados

exemplo SELECT ... WHERE nome LIKE '[A-Z]%'

intervalos

<attr> BETWEEN n AND y

verifica se atributo está dentro da gama de valores

exemplo SELECT ... WHERE (salary BETWEEN 630 AND 1000)

ordenação

ORDER BY <a1>, <a2> DESC, <a3> ...

obs ASC por defeito

quando realizamos
junção sobre tabelas
com o mesmo nome
devemos renomeá-las!

NULL

NULL + 5 = NULL

IS NULL verifica se atributo é nulo

IS NOT NULL

comparação com null retorna UNKNOWN
lógica de 3 valores

junções

SELECT ... FROM (<table1> JOIN <table2> ON <condition>) ...

junção de tabelas de acordo com condição

SELECT ... FROM (<table1> NATURAL JOIN <table2>) ...

não disponível
SQL Server

tabelas com mesmos atributos de junção, sendo os repetidos removidos
podemos renomear os atributos de relação para permitir junções

+ ... LEFT | RIGHT | FULL [OUTER] JOIN

obs! é possível o encadeamento de operações de junção

agregações

3

COUNT SUM MAX MIN AVG CONCAT

GROUP BY <grouping attributes>

HAVING <condition> → efetuar seleção sobre dados agrupados

exemplos

SELECT SUM(salary) AVG(salary) FROM Employees

SELECT COUNT(*) FROM Employees

SELECT COUNT(DISTINCT salary) FROM Employees

SELECT -- FROM ... GROUP BY Department HAVING COUNT(*) > 2

no SQL

subconsultas

usar o resultado de uma query numa query

FROM | cálculo de relações auxiliares

WHERE | testes de pertença/comparação/cardinalidade de conjuntos

WHERE <attr1> IN (SELECT <attr2> FROM <table>)

→ pertença a conjunto IN NOT IN

WHERE <attr1> > ANY (...)

< ALL (...)

→ seleciona os tuplos cujos atributos sejam =, >, <, < do que pelo menos um tuplo (ANY) ou todos os tuplos (ALL)

obs = ANY é o mesmo que IN

? diferente!

WHERE NOT EXISTS (...)

seleciona elementos da relação que não constam no subquery

WHERE UNIQUE (--) não disponível em SQL Server

obtem elementos da relação que constam uma única vez no subquery

Subconsultas
correlacionadas

Quando a query interior depende de dados que lhe são fornecidos pela query exterior, esta é executada uma vez para cada resultado do SELECT exterior