

SQL

CISC437/637, Lecture #7

Ben Carterette

Copyright © Ben Carterette

1

SQL Queries

- Basic form of a SQL query:

```
SELECT [DISTINCT] target  
FROM relations  
WHERE qualification
```

- *target* is a list of attributes/fields
- *relations* is a list of relations to get data from
- *qualification* is a set of comparisons joined with AND/OR/NOT
- DISTINCT is an optional keyword for dropping duplicates

Copyright © Ben Carterette

2

Conceptual Evaluation

- A simple evaluation strategy for SQL queries:
 - Compute cross product of *relations*
 - Discard records that do not match *qualification*
 - Drop attributes not in the *target*
 - If DISTINCT is specified, drop duplicates
- This works for any basic query
 - But it is usually the least efficient way to evaluate a query
 - Query optimization finds better ways to compute the same answer

Copyright © Ben Carter@Co

3

Projection in SQL: SELECT

SELECT A1, A2, ...
FROM Relation

- Use SELECT * to skip projection (get all fields)
- Perform generalized projection on numeric fields by including arithmetic expressions or functions
 - SELECT c+A1 from Relation
 - SELECT A1*A2 from Relation
- String functions can be applied to character fields
 - SELECT upper(A1) from Relation
 - SELECT substring(A1, a, b) from Relation

Copyright © Ben Carter@Co

4

Selection in SQL: WHERE

```
SELECT A1, A2, ...
FROM Relation
WHERE P
```

- P is a sentence of comparison operators on fields A1, A2, ... and constants, linked with AND, OR, NOT
- Other useful keywords:
 - A1 BETWEEN c AND d is equivalent to A1 >= c AND A1 <= d
 - A1 LIKE 'str%' → match strings starting with "str"

Copyright © Ben Carter@Co

5

Cross-Product in SQL: FROM

```
SELECT R1.A1, R1.A2, ..., R2.B1, R2.B2, ...
FROM Relation1 R1, Relation2 R2, ...
WHERE P
```

- Usually you'd want a natural join
 - Specify WHERE R1.foreignKey = R2.primaryKey

Copyright © Ben Carter@Co

6