Combining outputs of queries

- (select …) union (select …)
- (select …) intersect (select …)
- (select ...) except (select …)

- Use at the top-level query or in subqueries

- The columns produced by the pair of select statements must be the same.

Union example

- (select name from person where salary >= 30000) union (select name from person where age <= 20);
- Could also be done with a “where” clause

  select name from person where (salary >= 30000) or (age <= 20);
- “union” often clearer when the where conditions become complex

DALHOUSIE UNIVERSITY *Inspiring Minds*

Intersection in mysql

- Intersection doesn't exist in mysql
- Simulate in mysql using inner join + disctinct:

  select distinct <column list> from <t1> join <t2> on <join criteria>

- Simulate in mysql using where..in clause:

  select id from t1 where t1.id in (select id from t2);

- Select id from t1 join t2 using (id)

DALHOUSIE UNIVERSITY
Inspiring Minds

Intersection Example

- (select name from person where salary >= 30000) intersect (select name from person where age > 20);

- Simulate in mysql using inner join + distinct:

  select distinct name from (select name from person where salary >= 30000) as s1 join (select name from person where age > 20) as s2 on s1.name = s2.name

Except / minus in mysql

- Except or minus keywords don't exist in mysql

- Simulate in mysql using left join

  select t1.id from t1 left join t2 on t1.id = t2.id where t2.id is null;

Except example

- (select name from person where salary >= 30000) except (select name from person where age > 20);

- Simulate in mysql using left join

```
select name from
  (select name from person where salary >= 30000) as s1
left join (select name from person where age > 20) as s2
on s1.name = s2.name where s2.name is null;
```

Views

- A view creates an abstraction of rows from one (or more) tables
  - Can be all rows or a subset of them
- Allows queries to use the view like a table
  - Shortens the syntax of some tables
  - Allows re-use of common table joins and restrictions
  - Allows individuals to see only the data that is relevant to them (or permitted for them to see).

DALHOUSIE UNIVERSITY *Inspiring Minds*

View syntax

- Create view <viewName> as <select statement> [with check option]

- <viewName> can then be used as a table in queries.
- Including the “with check option” designation means that any updates requested through the view will check the where statement clauses before happening

- Delete the view with

  drop view <viewName>

View example
Using the sales database from last week's lab

- create view London_Employees as select * from employees where officeCode = 7;

  select * from London_Employees;

  drop view London_Employees;

  create view NA_Employees as select employees.*, territory from employees natural join offices where officeCode in (select officeCode from offices where territory = "NA");

  select * from NA_Employees;

  drop view NA_Employees;

DALHOUSIE UNIVERSITY *Inspiring Minds*

View example

- What if I wanted to create a view where an employee only sees the employee records of people in the same territory as them?
  - Helper: user() is a function that returns the login name of the individual who is running the query.

DALHOUSIE UNIVERSITY
Inspiring Minds

Just the tip of the SQL iceberg

- Other functionality to be aware of:
  - Case statements
    - Allows ”if…then” functionality in queries to change behavior
  - Variables
    - Set @var = <expression>
    - Select @var := <column> from …
  - For/while/repeat statements
    - Allows looping over the results of a query within SQL
  - With statements
    - Allows you to pull subqueries out of the main query and to not repeat the subquery text

Just the tip of the SQL iceberg

- **Other functionality to be aware of:**
  - **Stored procedures**
    - **Keeps a sequence of SQL commands in the DBMS that you can invoke with one command**
      - Gives flexibility, efficiency, shareability, applicability to more than one database
  - **Triggers**
    - **SQL to run before, after, or replacing specific commands to the database**

Case statement

- Format case [when…then…]+ [else …] end
- Example
  - Select city, case when territory = "NA" then "North America" else territory end as Territory from offices;

DALHOUSIE UNIVERSITY
Inspiring Minds

With statement example

- **Represent**

```
select EmployeeID, FirstName, LastName
  from employees
  where EmployeeID in (select distinct ReportsTo from employees);
```

as

```
with supervisorIDs as
  (select distinct ReportsTo from employees )

select EmployeeID, FirstName, LastName
  from employees
  where EmployeeID in supervisorIDs;
```

Changing records

- Use the “update” command:
  - Update <tablename> set [<column>=<value>]+ where …
  - Can set the value of multiple columns at the same time
  - Same “where” understanding as in select
    - Can use select subqueries to give a list
  - Values to set can be relative to the current value
    - Use the column name in the value clause
    - Will vary by row matched

DALHOUSIE UNIVERSITY
Inspiring Minds

Removing records

- Use the “delete” command:
  - Delete from <tablename> where …
  - Same “where” understanding as in select
    - Can use select subqueries to give a list

CRUD operations

- Create
  - Insert into … values …
- Read
  - Select … from … where …
- Update
  - Update … set … where …
- Delete
  - Delete from … where …

DALHOUSIE UNIVERSITY *Inspiring Minds*

Effect of timing

- By default, MySQL operates in “auto commit” mode
  - Each statement is stored in the database as you write it.
- There may be times when you need 2 (or more) statements to be done together or not at all to avoid conflicting information in the database:
  - The two updates might both be needed, but others may be changing the database at the same time as you
    - Eg. Change provincial and federal sales tax at the same time
      - Don’t want an invoice with inconsistent tax levels
  - If the second statement fails then you don’t want the first statement done
  - You’re trying out a change and may want to discard it if the process isn’t as you expected.

DALHOUSIE UNIVERSITY *Inspiring Minds*

Transactions

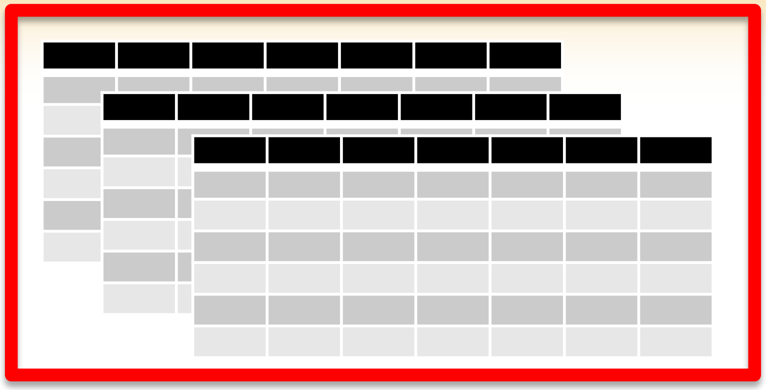
- A transaction is a construct where all SQL commands in the transaction are either have all done or have none done
  - Need to take the database out of ”auto commit” mode
- Identify the start and end of the group of statements
  - Start:
    - Start transaction
  - End:
    - Commit – put all the outputs into the database
    - Rollback – discard all the work of the transaction

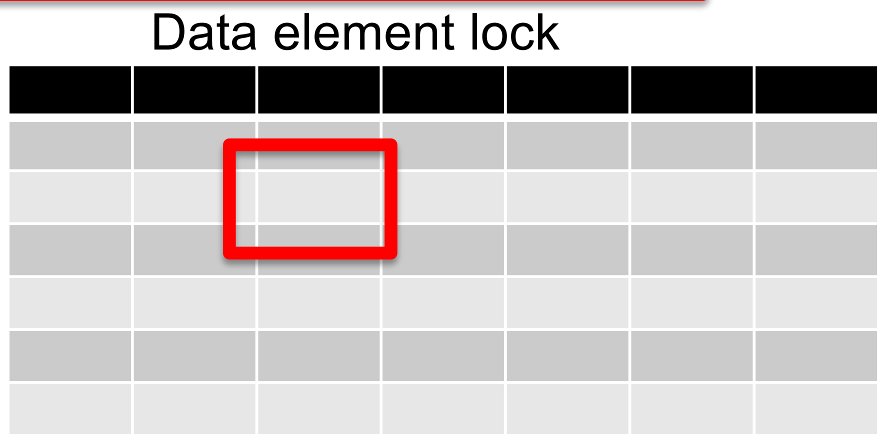
DALHOUSIE UNIVERSITY *Inspiring Minds*

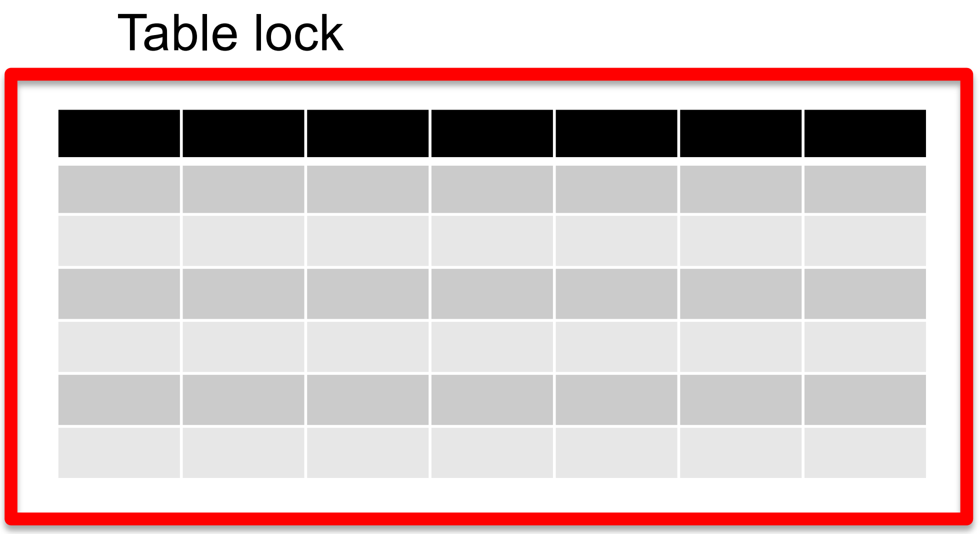
ACID properties – key for a DBMS to maintain

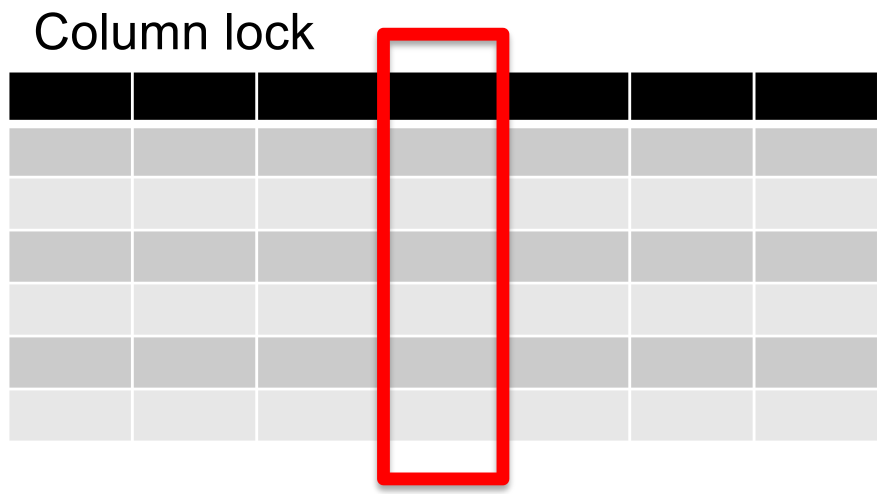
- Atomic
  - The transaction cannot be subdivided. It is either complete done or no part is done.
- Consistent
  - Any database constraint / property / relation that existed before the transaction must also exist after the transaction
- Isolated
  - Changes to the database are not revealed to users until the transaction is committed
- Durable
  - Changes are permanent

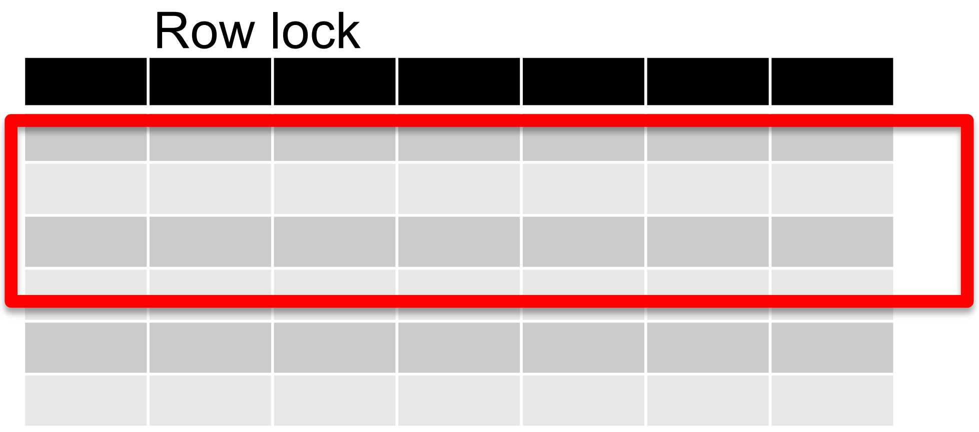
Locking











Lock granularity