Same as in the Quercus portion, we will use the following schema for a database about a company's employees:

- employee(<u>eid</u>, name, salary, dept)
  A tuple employee(a, b, c, d) indicates that employee a, whose name is b, has salary c and works in department d.

- department(<u>did</u>, name, division)
  A tuple department(a, b, c) indicates that department a has name b and is in division c. A division is a group of departments in the company.

- sales(<u>eid, day</u>, amount)
  eid and day together form a key. A tuple sales(a, b, c) indicates that employee a had sales valued at c dollars on day b.

- manages(<u>manager, junior</u>)
  manager and junior together form a key. A tuple manages(a, b) indicates that employee a manages employee b.

- employee[dept] $\subseteq$ department[did]

- manages[manager] $\subseteq$ employee[eid]

- manages[junior] $\subseteq$ employee[eid]

- sales[eid] $\subseteq$ employee[eid]

Throughout these prep exercises, and all our work in relational algebra, assume that every attribute in every tuple has a value. For those of you who happen to know some SQL, this means that there are no null values.

Important: Look closely at the integrity constraints; they will help you put tables together in the appropriate way. Notice that the attribute name on the LHS of a referential integrity constraint does not have to be the same as the attribute name on the RHS.

Important: The MarkUs syntax for relational algebra is finicky. Please read the four points about it on the Preps page on Quercus. These cover some specific things that past students have gotten snagged on.

You are about to write relational algebra queries on this schema. So far, we have learned these operators:

- select, expressed as `\select`

- project, expressed as `\project`

- Cartesian product, expressed as `\product`,

- natural join, expressed as `\natural_join`, and

- the set operators: union (`\union`), intersection (`\intersect`), and set difference (`\difference`).

You need nothing more than these operators to solve these questions.

You are welcome to use assignment, as we have done in class, to break an answer down into steps. Just make sure that the final step is an expression that yields the answer when evaluated. It should not be an assignment.

The full documentation on MarkUs syntax for relational algebra is available on the Preps page on Quercus.

How to Submit

Enter your answers into **prep3.txt**: your answers should all be placed under the line of `---`s. For example, your answers should look as follows:

```
================================================================================

[Q1] The question will be asked here.

--------------------------------------------------------------------------------

ENTER YOUR ANSWER HERE

================================================================================
```

You may run the automated self-tests on MarkUs in order to check the correctness of your answers. To do this, submit your prep3.txt and then click on the "Automated Testing" tab. Click "Run tests": your results should appear after a few minutes.

Employee Schema

employee(eid, name, salary, dept)

department(did, name, division)

sales(eid, day, amount)

manages(manager, junior)

employee[dept] ⊆ department[did]

manages[manager] ⊆ employee[eid]

manages[junior] ⊆ employee[eid]

sales[eid] ⊆ employee[eid]