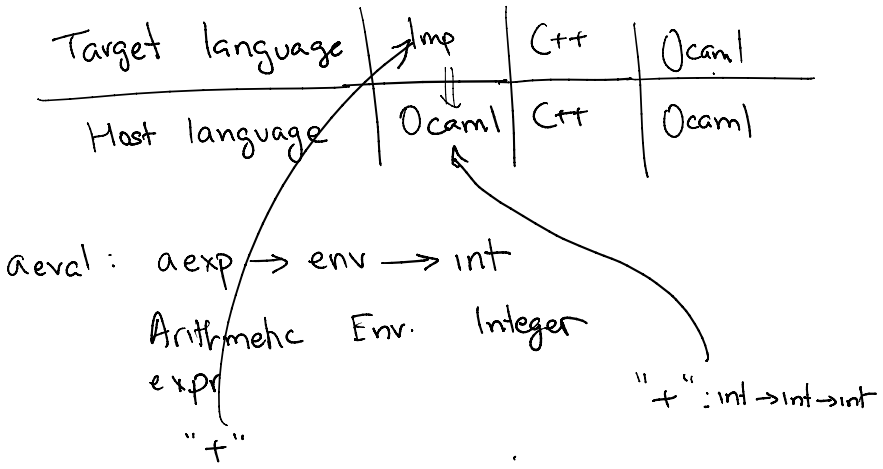
Lecture 21

Tuesday, November 3, 2020 4:05 PM

Question: Implement ceval : cmd → int list



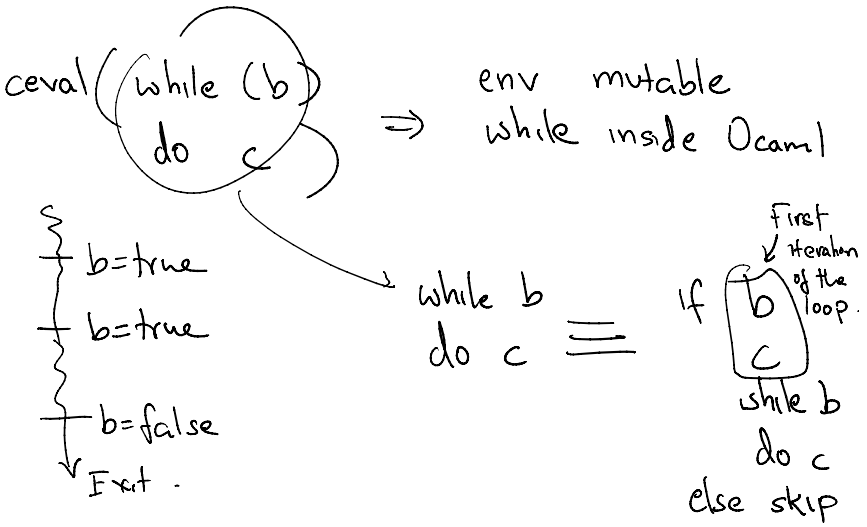
$\overline{\text{aeval}\ \overline{(e_1 + e_2)}} = (\text{aeval}\ e_1) \oplus (\text{aeval}\ e_2)$

aeval Plus(e_1, e_2) $= (\text{aeval}\ e_1) + (\text{aeval}\ e_2)$

aeval Minus(e_1, e_2) $= (\text{aeval}\ e_1) - (\text{aeval}\ e_2)$

ceval If$(e_1\ e_2\ e_3)$ = if beval e_1 then ceval e_2 else ceval e_3

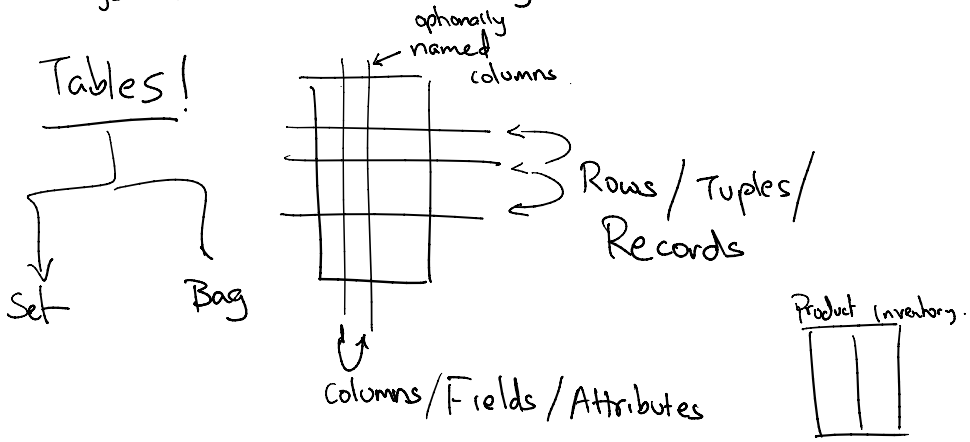
Metalinguistic abstraction



output $\overline{1 - \overline{-1}}$

SQL / Relational algebra

Things that we are talking about:



(int * int * string) : single row

(int * int * string) set : entire table

Basic types: null int real text blob.

$$\begin{bmatrix}1\\2\\4\end{bmatrix} \cup \begin{bmatrix}3\\2\\8\end{bmatrix} \Rightarrow \begin{bmatrix}1\\2\\4\\3\\2\\8\end{bmatrix}$$ Duplicate?

$$\begin{bmatrix}1\\2\\4\\3\\8\end{bmatrix}$$ Deduplicate

$$\left\{\begin{matrix}(1,1)\\(2,1)\\(4,1)\end{matrix}\right\} \cup \left\{\begin{matrix}(3,1)\\(2,1)\\(8,1)\end{matrix}\right\} \Rightarrow \left\{\begin{matrix}(1,1)\\(2,2)\\3,1)\\(4,1)\\(8,1)\end{matrix}\right\}$$

$\{1, 2\} \cup \{3, 2\} = \{1, 2, 3\}$ (In sets)

$\{1, 2\} \cup \{3, 2\} = \{1, 2, 3\}$ (In sets)

$\{1, 2\} \cup \{3, 2\} = \{1 \underset{2}{2} 3\}$

$\{1 \underset{2}{2} 3\} \cap \{\underset{5}{1} \underset{8}{2}\} = \{\underset{1}{1} \underset{2}{2} \underset{0}{3}\}$

$\{1\ 2\ 3\} \cap \{1\ 2\} = \{1\ 2\}$

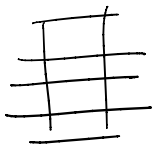
Table = set of tuples

$Q ::= T$ (just a table)

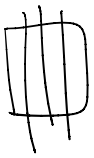
$\mid Q_1 \cup Q_2$ (union)

$\mid Q_1 \cap Q_2$ (intersection)

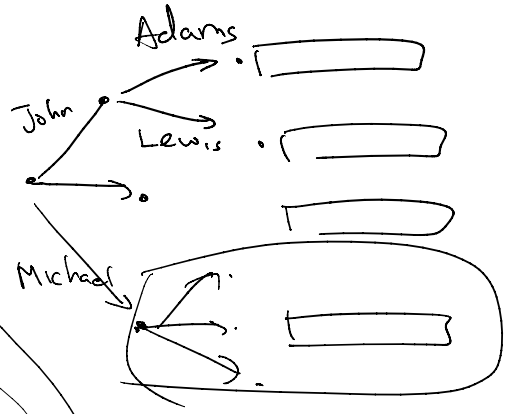
$\mid Q_1 \setminus Q_2$ (difference)



$\mid \sigma_{\text{filter}} Q$ (select) $\sigma_{\text{Name = Michael}}$ Employees



$\mid \pi_{\text{columns}} Q$ (project)

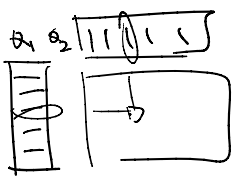


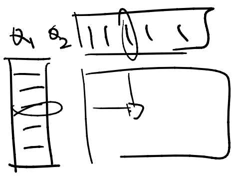
$\mid Q_1 \bowtie Q_2$ (join)

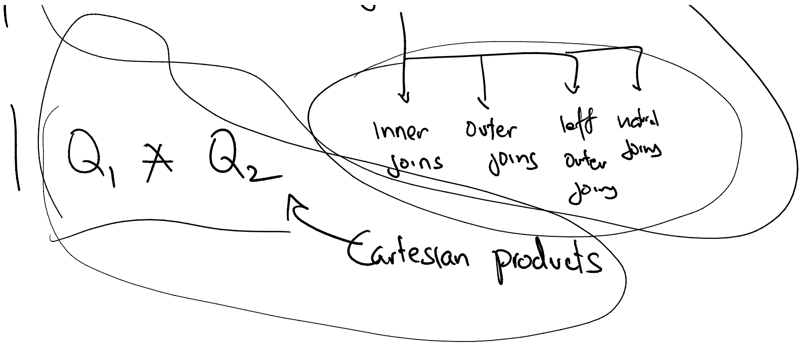
$\mid Q_1 * Q_2$

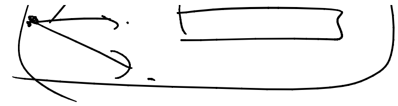
Inner joins, Outer joins, left outer joins, right outer joins

Cartesian products





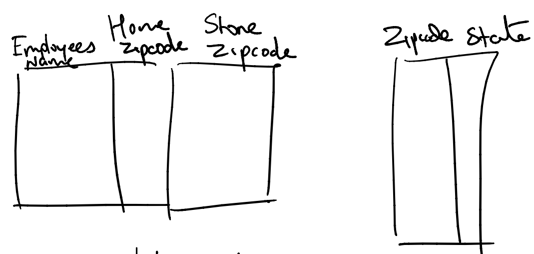




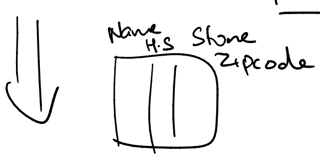
$\sigma_{(dob=today)}$ Employees = "All employees whose birthday is today"

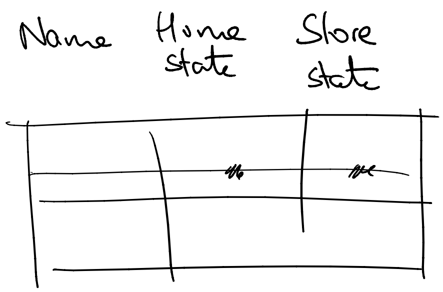
$\sigma_{salary>90K \text{ and daily commute} < 40 min.}$ Employees

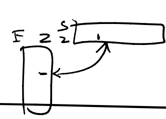
$\pi_{hometown}$ Employees.



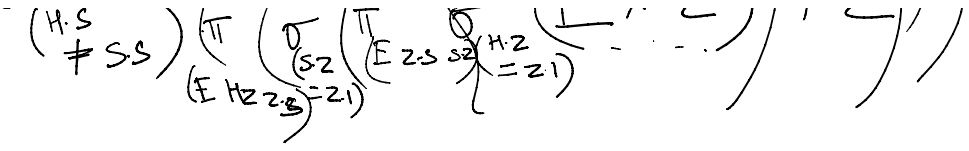
Which employees live in one state but commute to another?







$\sigma_{(H.S \neq S.S)}\left(\pi_{(E\ HZ\ Z \ldots = Z.1)}\left(\sigma_{(S.Z}\left(\pi_{(E\ Z.S\ S.Z)}\left(\sigma_{(H.Z = Z.1)}(E \times Z)\right) \times Z\right)\right)\right)$



let $T_1 = \pi_{E.1,\ z.2,\ E.3}\ \sigma_{E.2 = z.1}(E * z)$

E.1 → Name; z.2 → Home state; E.3 → Store zip code

let $T_2 = \pi_{T_1.1,\ T_1.2,\ z.2}\ \sigma_{T_1.3 = z.1}(T_1 * z)$

$T_1.1$ → Employee name; $T_1.2$ → Home state; z.2 → Store state

Final result = $\sigma_{T_2.2 \neq T_2.3}\ T_2$

$T_2.2$ → Home state; $T_2.3$ → store state.

SQL where

$\sigma_{E.2 = z.1}(E * z) = E \bowtie_{E.2 = z.1} z$

Group by: Outside the bounds of traditional RA.

```
select Employee.LastName, Employee.FirstName,
Employee.Email
from Employee
where Employee.HireDate > "2003-01-01"

/* Question: Is the following a valid translation of the
SQL query into R.A?
   Project (Lname, Fname, Email) (Select (HireDate > 2003)
Employee) */
```

```
select E1.LastName
from Employee as E1
where exists (select * from Customer as E2 where E1.LastName = E2.LastName)

// project (Employee.LastName) select (Employee.LastName = Customer.LastName) Employee *
```

Customer

// select in SQL ~= project in RA
// from in SQL ~= Cartesian Product / Join in RA
// where in SQL ~= select in RA

```
select Employee.Title, count(*)
from Employee
group by Employee.Title

// Group the entries of the Employee table by their title
// Count the number of employees in each group

// Group the employees by their title
// Count the number of employees having each title
```