SEMANTIC OPERATIONS
introduction

XPLAIN DATA MANIPULATION LANGUAGE COMMANDS:

- **SELECTION**

  based on the inherent constraints in the data model.

- **EXTENSION**

  necessary to derive information;
  crucial element in the formulation of complex queries.

- **MODIFICATION**

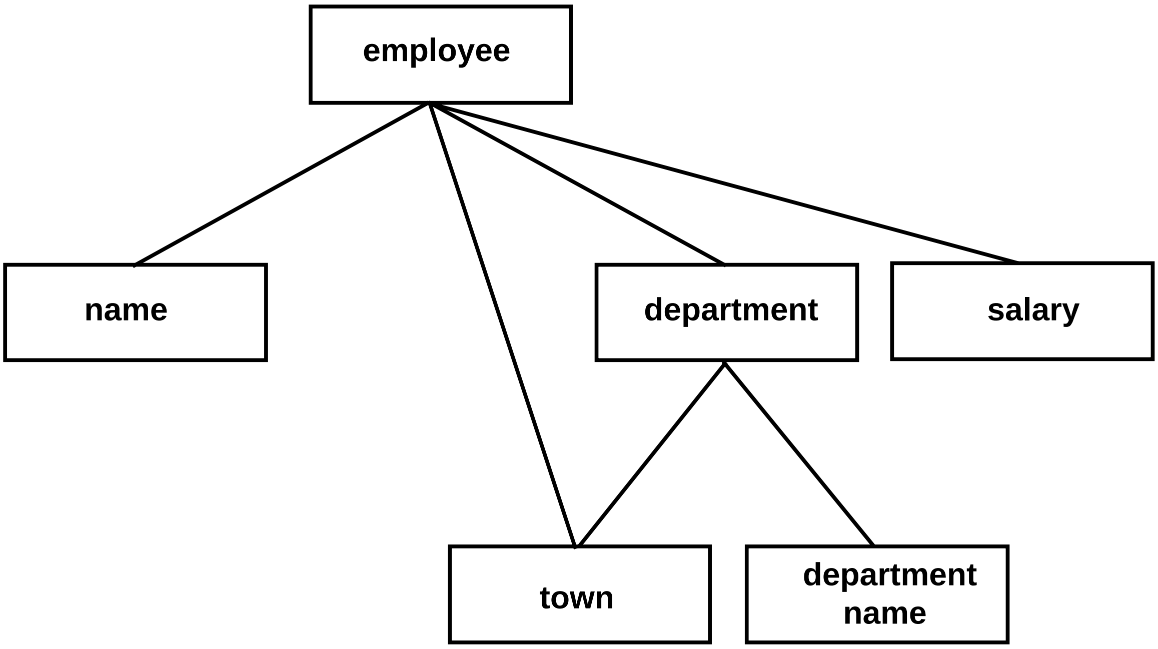
  commands needed to change the contents of a database.

EVALUATION:

- **the orthogonality principle;**
- **comparison with relational languages.**

SELECTION

type **department = department name, business_town**
type **employee = name, home_town, department, salary.**



Example 1: Select data of the employee with the identification E3.

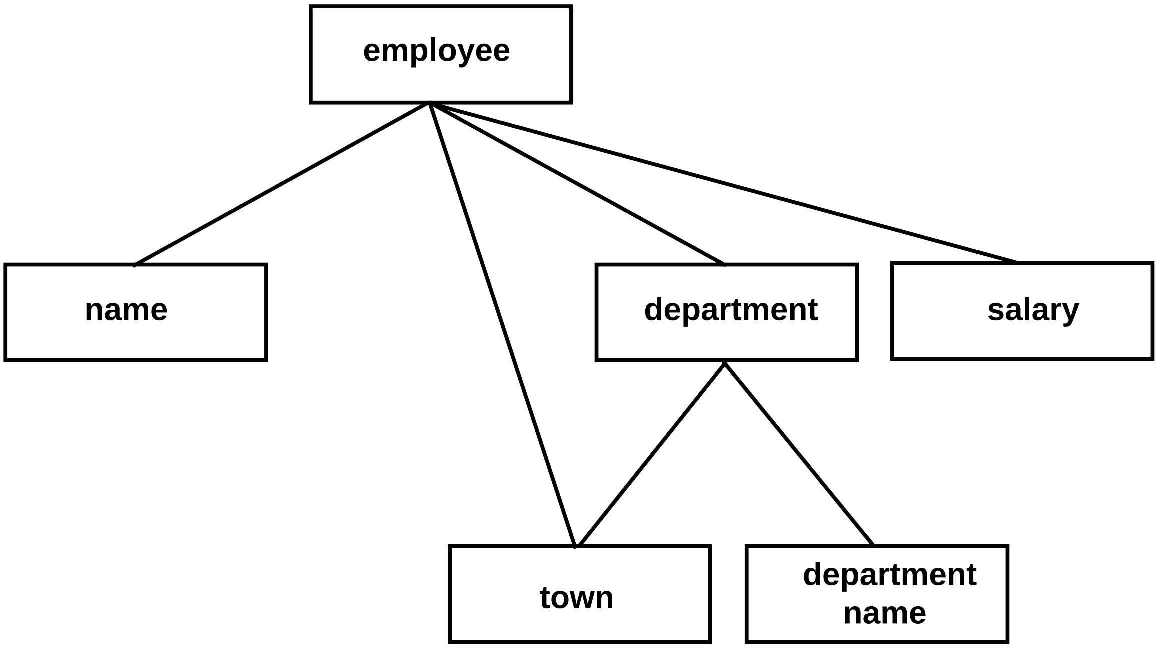
get **employee "E3".**

Example 2: Select employees living in Guilding.

get **employee** ***its*** **name, department**
    where **home_town = "Guilding".**

SELECTION
(continued)

type **department = department name, business_town**
type **employee = name, home_town, department, salary.**



Example 3: Select commuters.

get **employee** ***its*** **name, home_town, department**
 where **home_town ≠ department** ***its*** **business_town.**

Example 4: How many employees work in Guilding?

get count **employee**
 where **department** ***its*** **business_town = "Guilding".**

SET FUNCTIONS

The following functions are available:

- ***count*: returns the number of elements in a set;**
- ***max*: returns the maximum value of a set of attribute values;**
- ***min*: returns the minimum value of a set of attribute values;**
- ***total*: returns the sum of the values of a set of attribute values;**
- ***nil*: returns ’true’ if the selected set equals the empty set, otherwise ’false’;**
- ***any*: returns ’false’ if the selected set equals the empty set, otherwise ’true’;**
- ***some*: returns a random element of the selected set.**

Example 6: What is the highest salary?

***get max* employee *its* salary.**

Example 7: Are there any employees earning more than 50,000?

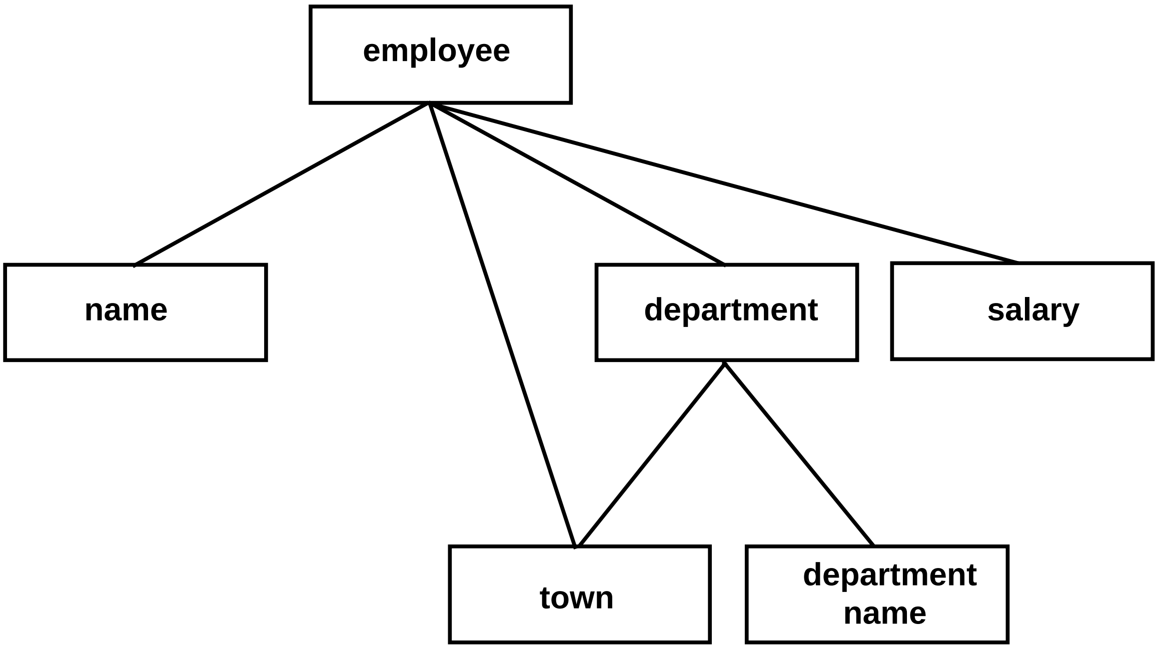
```
get any employee
    where salary > 50000.
```

Example 8: Select the name of an arbitrary employee in the Purchase department.

```
get some employee its name
    where department its department name =
    "Purchase".
```

EXTENSIONS

type **department = department name, business_town**
type **employee = name, home_town, department, salary.**



Example 9: Provide an overview of the departments, including the number of employees.

extend **department** ***with*** **number of employees =**
count **employee**
per **department.**

get **department** ***its*** **department name, business_town, number of employees.**

EXTENSION RESULTS IN GENERIC SOLUTIONS, IT CAN BE USED FOR SEVERAL PURPOSES !!

EXTENSION
(continued)

```
type department   = department name, business_town
type employee     = name, home_town, department,
                    salary.
```

Example 10: Select departments with more than 100 employees.

(FIRST STEP OF EXAMPLE 9: number of employees)

```
get department its department name, business_town
    where number of employees > 100.
```

Example 11: Find the number of departments with more than 100 employees.

(FIRST STEP OF EXAMPLE 9: number of employees)

```
    get count department
        where number of employees > 100.
```

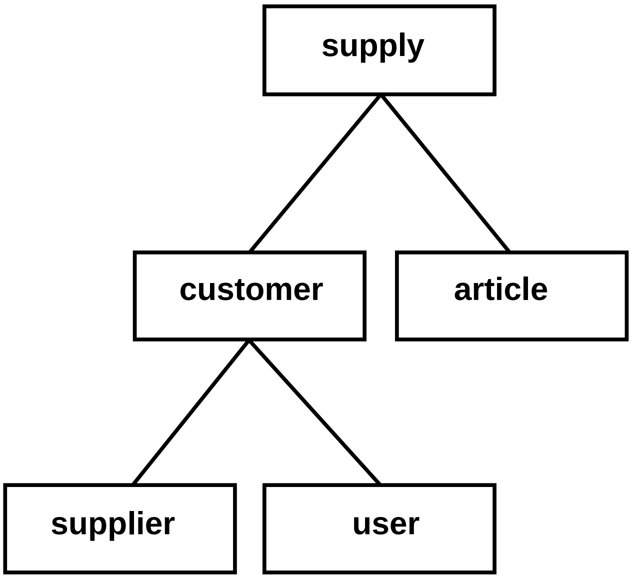
Example 12: Which department has the most employees?

(FIRST STEP OF EXAMPLE 9: number of employees)

```
value maximum =
    max department its number of employees.

get department its department name, business_town
    where number of employees = maximum.
```

EXTENSION
(continued)

type **supplier = company name, business_town**
type **article = make, price**
type **user = user name, home_town**
type **customer = supplier, user**
type **supply = customer, article, quantity,**



Example 13: Users being customers of all suppliers?

PSEUDO: {customer *its* supplier *per* user} = {supplier},

```
extend user with number =
      count customer
      per user.

value number of suppliers =
      count supplier.

get user its user name, home_town
      where number = number of suppliers.
```

EXTENSION
(continued)

Example 14: Select suppliers with the same users as customer as the supplier identified by S3.

PSEUDO: {customer *its* user *per* supplier}

$\supseteq$

{customer *its* user *where* supplier = "S3"}.

$\Downarrow$

{{customer *its* user *per* supplier} $\cap$ {customer *its* user *where* supplier = "S3"}}

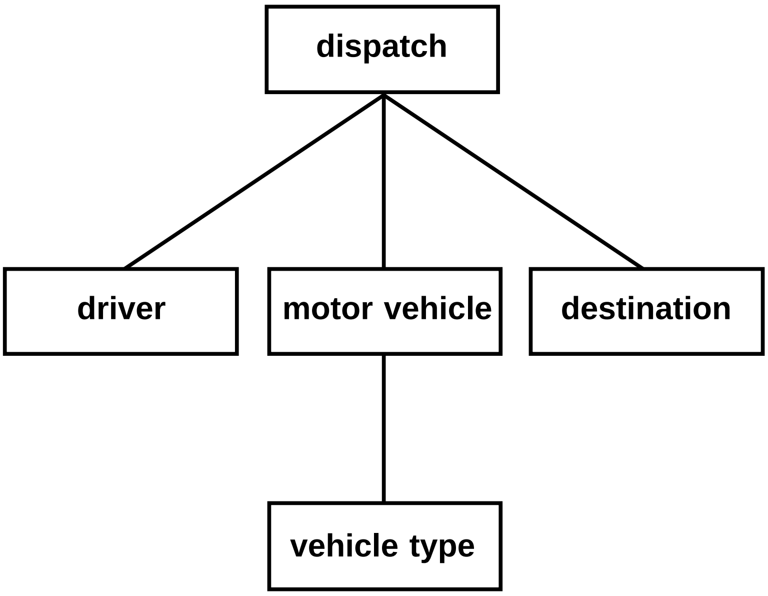
=

{customer *its* user *where* supplier = "S3"}.

```
extend user with S3 customer =
        any customer
        where supplier = "S3"
        per user.

extend supplier with number of S3 customers =
        count customer
        where user its S3 customer
        per supplier.

value number S3 =
        count user
        where S3 customer.

get supplier its company name, business_town
        where number S3 = number of S3 customers.
```

EXTENSION
(continued)

type dispatch = driver, motor vehicle, destination, cargo
type driver = name, address, home_town
type destination = company, address, town
type motor vehicle = make, vehicle type, year, chassis number
type vehicle type = manufacturer, engine capacity, fuel.

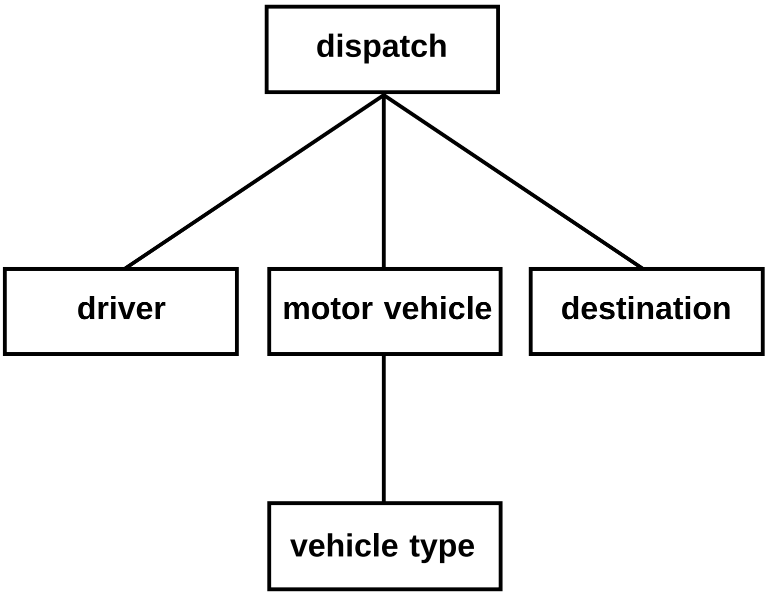


Example 16: Drivers operating on all destinations.

```
extend driver with number =
    count dispatch its destination
    per driver.

value number of destinations =
    count destination.

get driver
    where number = number of destinations.
```

EXTENSION
(continued)

type dispatch = driver, motor vehicle, destination, cargo
type driver = name, address, home_town
type destination = company, address, town
type motor vehicle = make, vehicle type, year, chassis number
type vehicle type = manufacturer, engine capacity, fuel.



Example 17: Which vehicle type is most frequently used in dispatches?

extend vehicle type *with* number =
  count dispatch
  per motor vehicle *its* vehicle type.

value maximum =
  max vehicle type *its* number.

get vehicle type *where* number = maximum.

MODIFICATION

```
type employee = name, home_town, department, salary.
```

- **INSERT**

Example 18: Insert an employee with the attributes identification (E20), name (Fletcher), home_town (Guilding), department (D3) and salary (15,000).

```
insert employee "E20"
    its name = "Fletcher",
    home_town = "Guilding",
    department = "D3",
    salary = 15000.
```

- **DELETE**

Example 19: Remove all employees whose salary is zero.

```
delete employee
    where salary = 0.
```

- **UPDATE**

Example 20: Increase the salaries of employees earning less than 50,000 by 5%.

```
update employee its salary = 1.05 * salary
    where salary < 50000.
```

EVALUATION

ORTHOGONALITY PRINCIPLE:

- the number of concepts should be as small as possible (though not necessarily minimal).
- a given syntactical construct should have the same meaning in various situations.
- given semantic constructs should have the same structure under all conditions.
- each command should be atomic.
- the use of query languages should not be subject to arbitrary restrictions.
- the execution of commands should not be associated with unexpected side-effects.
- nested expressions should be subject to as few restrictions as possible.
- a language must have been defined carefully.
- a language should enhance reproducibility.
- a language should be helpful in the retrieval of useful information.
- there should be few opportunities for the introduction of errors.
- a language should have efficient programming, translating (possibly interpreting) and processing capabilities.
- a language should support the model in use.

EXERCISES

TYPE DEFINITIONS:

type employee = name, address, home_town,
salary, manager_employee,
department
type department = name, floor
type article = description, class
type purchase article = wholesaler, article
type supply = purchase article, department,
quantity
type sales article = article, department
type sale = sales article, quantity
type wholesaler = name, address,
business_town.

EXERCISES:

5 Select employees working in the toys department (identification D20).

6 Select the articles sold on the second floor.

7 Select the articles not sold on the second floor.

8 Select the articles sold by all departments on the second floor.

9 Select the salary of the manager of employee identified by E32.

10 Select the employees earning more than their managers.

EXERCISES
(continued)

TYPE DEFINITIONS:

***type* employee = name, address, home_town, salary, manager_employee, department**

***type* department = name, floor**

***type* article = description, class**

***type* purchase article = wholesaler, article**

***type* supply = purchase article, department, quantity**

***type* sales article = article, department**

***type* sale = sales article, quantity**

***type* wholesaler = name, address, business_town.**

EXERCISES:

11 Select the departments where the average salary is in excess of 12,500.

12 Select employees earning more than any employee in the shoes department (D23).

13 Select the wholesalers providing pens.

14 Select the articles sold by all departments.

15 Select the articles supplied by only one wholesaler.

16 Select the wholesaler with the broadest range of articles.