1 Schema and views

There are many valid schemas here.

Here is the one I have:

```
                                  Table "public.possession"
 Column  |  Type   | Collation | Nullable | Default | Storage  | Stats target | Description
---------+---------+-----------+----------+---------+----------+--------------+-------------
 id      | integer |           | not null |         | plain    |              |
 type    | integer |           |          |         | plain    |              |
 owner   | integer |           |          |         | plain    |              |
 price   | integer |           |          |         | plain    |              |
Indexes:
    "possession_pkey" PRIMARY KEY, btree (id)
Foreign-key constraints:
    "possession_owner_fkey" FOREIGN KEY (owner) REFERENCES player(id)
    "possession_type_fkey" FOREIGN KEY (type) REFERENCES type(id)


                                  Table "public.player"
 Column |  Type   | Collation | Nullable | Default | Storage  | Stats target | Description
--------+---------+-----------+----------+---------+----------+--------------+-------------
 id     | integer |           | not null |         | plain    |              |
 name   | text    |           |          |         | extended |              |
 money  | integer |           |          |         | plain    |              |
Indexes:
    "player_pkey" PRIMARY KEY, btree (id)
Check constraints:
    "money_above_0" CHECK (money >= 0)
Referenced by:
    TABLE "possession" CONSTRAINT "possession_owner_fkey" FOREIGN KEY (owner) REFERENCES player(id)


                                   Table "public.type"
 Column |  Type   | Collation | Nullable | Default | Storage  | Stats target | Description
--------+---------+-----------+----------+---------+----------+--------------+-------------
 id     | integer |           | not null |         | plain    |              |
 name   | text    |           |          |         | extended |              |
Indexes:
    "type_pkey" PRIMARY KEY, btree (id)
Referenced by:
    TABLE "possession" CONSTRAINT "possession_type_fkey" FOREIGN KEY (type) REFERENCES type(id)
```

We could also have decided that id are serial which means that postgres would automatically generate new unique id. The NULL in price means that the object is not buyable.

Now for the operations:

- Create a new type of object or add new players,

  ```
  INSERT INTO player (id,name,money) VALUES (<id>,<name>,<money>);
  ```

- Change the name of a type of objects or the name of a player,

  ```
  UPDATE player SET name = <name> WHERE id = <id>;
  ```

- Attribute an object of a given type to a player,

  ```
  INSERT INTO possession (type,owner,price)
    VALUES (<typeId>,<ownerId>,NULL) ; %mehwish: the id should be serial in the t
   -- the id is generated automatically!
  ```

- Increase or decrease the amount of money a player has,

```
UPDATE player SET money = money+<diff>;
```

- Retrieve the list of all the items that a player has,

```
SELECT type, owner FROM possession;
```

- Compute the current balance of a player,

```
SELECT money FROM player WHERE id = <id>;
```

- Allow a player to mark one of their item as buyable with a given price,

```
UPDATE possession SET price = <price> WHERE id = <id> ;
  -- we can also check that the player id is correct with
UPDATE possession SET price = <price> WHERE id = <id> AND owner = <playerId> ;
```

- Allow a player to buy the cheapest item of a given type from the marketplace.

```
START TRANSACTION
SELECT id,price,owner as curOwner
  FROM possession
  WHERE buyable IS NOT NULL AND type=<desired_type>
  ORDER BY price LIMIT 1 ; -- we get the price and the id
UPDATE player SET money = money + price WHERE id = curOwner ;
UPDATE player SET money = money - price WHERE id = <buyer> ;
UPDATE possession SET owner = <buyer> AND price = NULL WHERE id = idObject
COMMIT
  -- note that the price = NULL makes the object unbuyable
```

2 Normalization & Advanced Exercises

- Create a department whose employees are located in different buildings using multivalued attributes.

```
Table "public.department"
   Column    |  Type   | Collation | Nullable | Default
-------------+---------+-----------+----------+---------
 dnumber     | integer |           | not null |
 dname       | text    |           |          |
 d_head      | integer |           |          |
 d_building  | text    |           |          |
Indexes:
    "department_pkey" PRIMARY KEY, btree (dnumber)
```

- Retrieve information about a department based on the location,

```
SELECT * FROM department WHERE d_building LIKE '%BuildingB%';
SELECT * FROM department WHERE d_building = 'BuildingB';
```

- Normalize the department information to comply with 1NF.

```
Table "public.department_normalized"
   Column    |  Type   | Collation | Nullable |      Default
-------------+---------+-----------+----------+--------------------
 id          | integer |           | not null | nextval('department_normalized
                                                  _id_seq'::regclass)
 dnumber     | integer |           |          |
```

```
 dname      | text    |           |          |
 d_head     | integer |           |          |
 d_building | text    |           |          |
Indexes:
    "department_normalized_pkey" PRIMARY KEY, btree (id)
```

- Retrieve information about a department based on the location using exact match.

```
SELECT * FROM department_normalized WHERE d_building = 'BuildingB';
```

- Create courses taught by the professors and attended by the students.

```
Table "public.teach"
  Column   | Type | Collation | Nullable | Default
-----------+------+-----------+----------+---------
 student   | text |           |          |
 course    | text |           |          |
 professor | text |           |          |
```

- Define possible decompositions of the courses.
  Decomposition 1

```
Table "public.teach_1_1"
  Column   | Type | Collation | Nullable | Default
-----------+------+-----------+----------+---------
 student   | text |           | not null |
 professor | text |           | not null |
Indexes:
    "teach_1_1_pkey" PRIMARY KEY, btree (student, professor)
```

```
Table "public.teach_1_2"
 Column  | Type | Collation | Nullable | Default
---------+------+-----------+----------+---------
 student | text |           | not null |
 course  | text |           | not null |
Indexes:
    "teach_1_2_pkey" PRIMARY KEY, btree (student, course)
```

Decomposition 2

```
Table "public.teach_2_1"
  Column   | Type | Collation | Nullable | Default
-----------+------+-----------+----------+---------
 course    | text |           |          |
 professor | text |           | not null |
Indexes:
    "teach_2_1_pkey" PRIMARY KEY, btree (professor)

Table "public.teach_2_2"
 Column  | Type | Collation | Nullable | Default
```

```
---------+------+-----------+----------+---------
 course  | text |           | not null |
 student | text |           | not null |
Indexes:
    "teach_2_2_pkey" PRIMARY KEY, btree (course, student)
```

Decomposition 3

```
Table "public.teach_3_1"
  Column   | Type | Collation | Nullable | Default
-----------+------+-----------+----------+---------
 course    | text |           |          |
 professor | text |           | not null |
Indexes:
    "teach_3_1_pkey" PRIMARY KEY, btree (professor)

Table "public.teach_3_2"
  Column   | Type | Collation | Nullable | Default
-----------+------+-----------+----------+---------
 student   | text |           | not null |
 professor | text |           | not null |
Indexes:
    "teach_3_2_pkey" PRIMARY KEY, btree (student, professor)
```

- Reconstruct the courses taught by the professors and attended by the students.

```sql
SELECT course, professor, t1.student
FROM teach_1_1 AS t1, teach_1_2 AS t2
WHERE t1.student = t2.student;

SELECT t1.course, professor, student
FROM teach_2_1 AS t1, teach_2_2 AS t2
WHERE t1.course = t2.course;

SELECT  course, t1.professor, student
FROM teach_3_1 AS t1, teach_3_2 AS t2
WHERE t1.professor = t2.professor;
```

- Create information about specific employees by taking into account the properties of the employees in general.

```
Table "public.employee"
 Column |  Type   | Collation | Nullable |         Default
--------+---------+-----------+----------+--------------------------
 id     | integer |           | not null | nextval('employee_id_seq'
                                           ::regclass)
 name   | text    |           |          |
 salary | integer |           |          |
Indexes:
    "employee_pkey" PRIMARY KEY, btree (id)
```

```
Table "public.professor"
 Column |  Type   | Collation | Nullable |         Default
--------+---------+-----------+----------+---------------------------
 id     | integer |           | not null | nextval('employee_id_seq'
                                           ::regclass)
 name   | text    |           |          |
 salary | integer |           |          |
 pid    | integer |           | not null |
 field  | text    |           |          |
Indexes:
    "professor_pkey" PRIMARY KEY, btree (pid)
Inherits: employee

Table "public.secretary"
  Column  |  Type   | Collation | Nullable |        Default
----------+---------+-----------+----------+--------------------------
 id       | integer |           | not null | nextval('employee_id_seq'
                                            ::regclass)
 name     | text    |           |          |
 salary   | integer |           |          |
 sid      | integer |           | not null |
 building | text    |           |          |
Indexes:
    "secretary_pkey" PRIMARY KEY, btree (sid)
Inherits: employee
```

- Retrieve information about all the types of employees.

```
SELECT * FROM employee;
SELECT * FROM ONLY employee;
SELECT * FROM professor;
SELECT * FROM secretary;
```

- Create courses having dependent types of information which would not exist otherwise.

```
 Table "public.course"
 Column |  Type   | Collation | Nullable |      Default
--------+---------+-----------+----------+-------------------------
 id     | integer |           | not null | nextval('course_id_seq'
                                           ::regclass)
 name   | text    |           |          |
Indexes:
    "course_pkey" PRIMARY KEY, btree (id)

Table "public.session"
 Column |  Type   | Collation | Nullable | Default
--------+---------+-----------+----------+---------
 course | integer |           | not null |
 num    | integer |           | not null |
 name   | text    |           |          |
Indexes:
```

```
    "session_pkey" PRIMARY KEY, btree (course, num)
Foreign-key constraints:
    "session_course_fkey" FOREIGN KEY (course) REFERENCES course(id)
```