# EDABA Task 1

Jakub Tomkiewicz 300183

## 1. Textual description of a database

### 1.1 Aim

Database of a body leasing company that hires developers and lease them to other businesses.
To lease an employee to a client, lease contracts are made for a specified time and money. Each concluded contract can be reviewed by a client, employee and owner of the company. Employees get equipment (computer, phone or tablet) on which they can work. History of rented equipment is tracked.

### 1.2 Objects

**Clients**
- ID
- Name
- Tax number
- Street
- Flat number
- City
- Country
- ZIP Code

**Lease contracts**
- ID
- Employee
- Client
- Price
- Date begin
- Date end

**Programming languages**
- ID
- Name

**Employees**
- ID
- Name
- Surname
- Personal ID Number
- Bank account
- Email
- Telephone
- Street
- Flat number
- City
- Country
- ZIP Code
- Programming language
- Earnings
- Cooperation date begin
- Archival

**Reviewers dictionary**
- ID
- Name

**Equipment**
- ID
- Name
- Specification
- Condition

**Equipment rental**
- ID
- Equipment
- Employee
- Date begin
- Date end
- Date of actual return

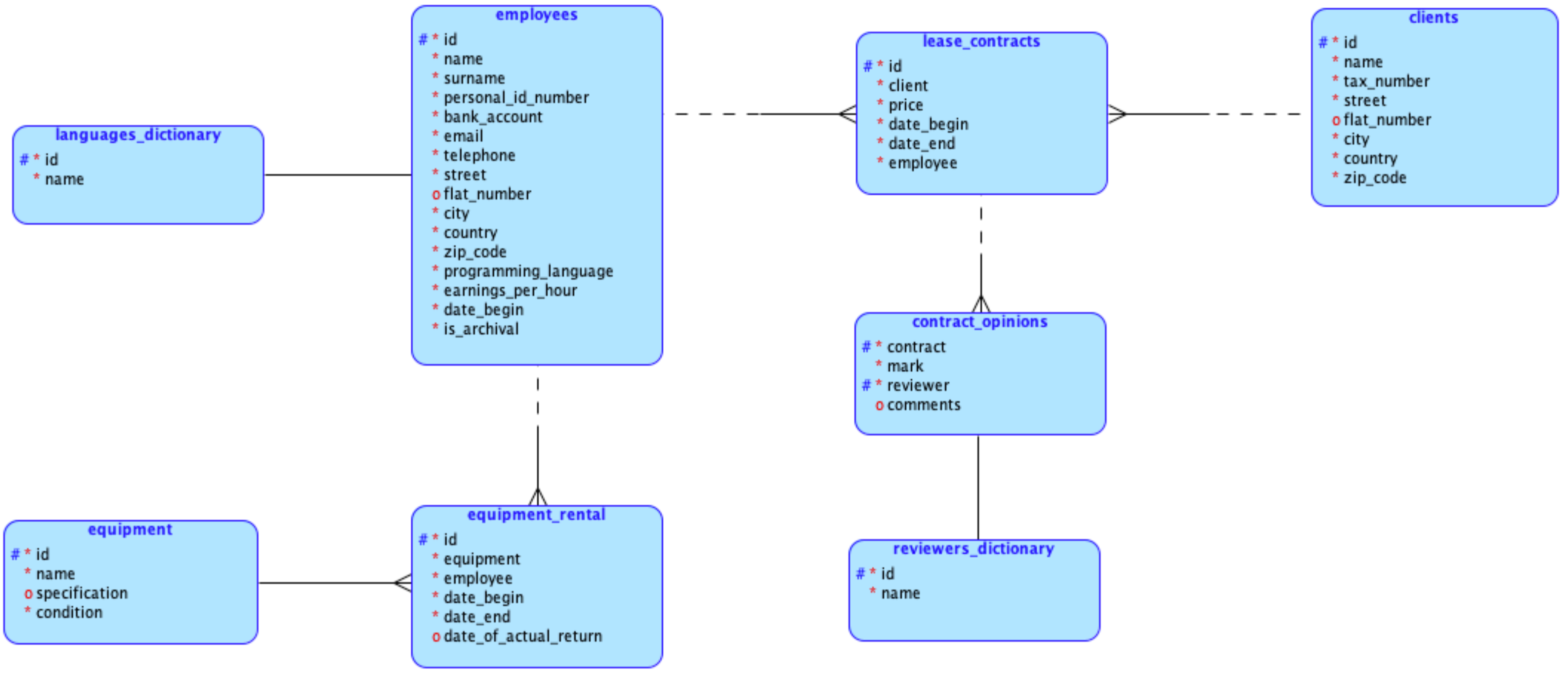
**Contract opinions**
- Contract
- Reviewer
- Mark
- Comments

### 1.3 Requirements concerning data

IDs, numbers and indexes are stored as integers. Numbers regarding money are doubles with 2 decimal places. All kinds of names and descriptions are varchars of various length. However, for very long variables text is used. Date format is applied for dates and boolean for true/false variables.
Most of the data is mandatory, but there are three values (flat_number, comments, date_of_actual_return) that do not have to be inserted.

# 2. ER diagram

## 2.1 Diagram



Data model created in Oracle SQL Developer.

## 2.2 Text description of entities and relations

Employees entity stores all personal data regarding employees with their contact information, date of the beginning of cooperation and address. Archive value is as default false, but for archive workers is true, in order to not delete their data. Programming language is stored as dictionary in separate entity connected with foreign key.

Clients entity stores names, tax numbers and addresses of clients, with whom we have cooperated so far.

Lease contracts connects Employees with Clients. It is concluded when employee is being leased for a client. Entity stores information about the costs, start and end dates of the lease.

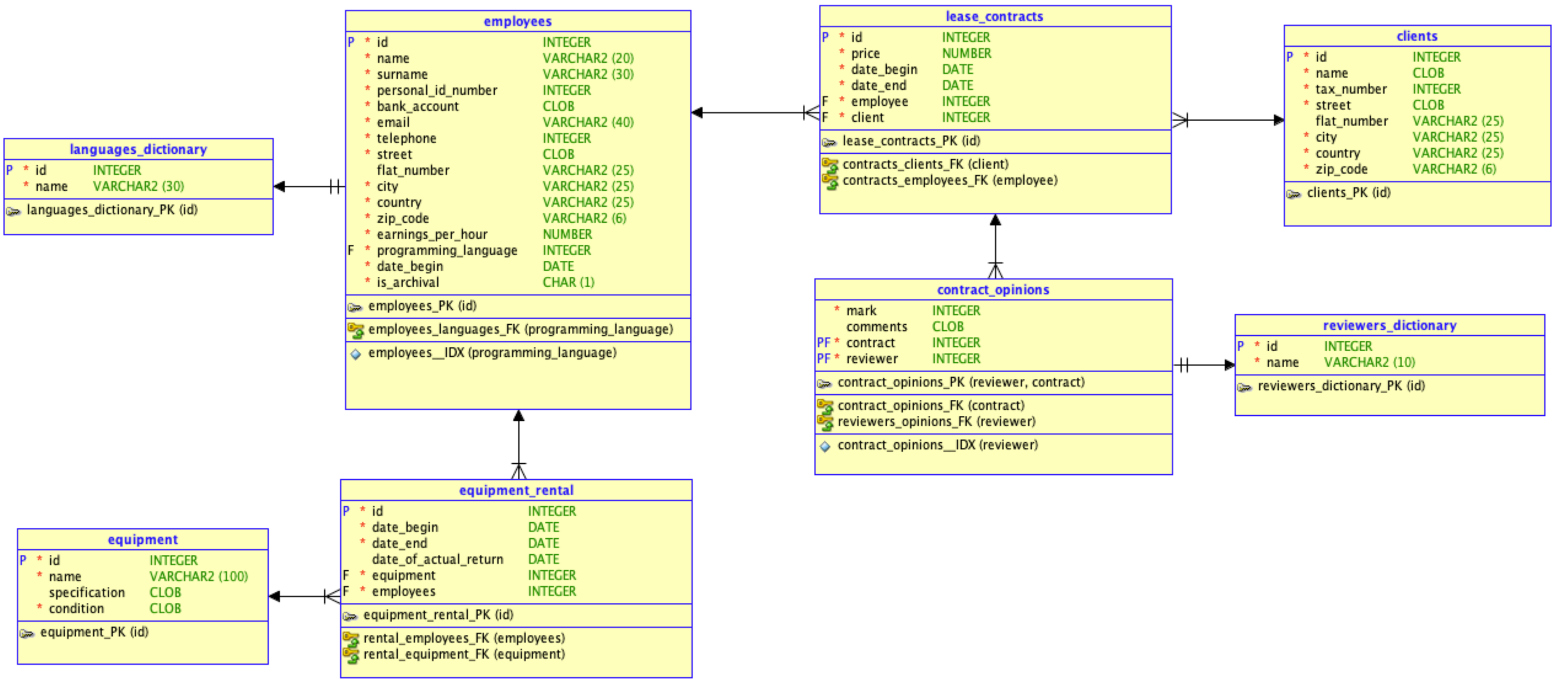
Using a foreign key Contract opinions entity is connected to Lease contracts. It stores marks (from 1 to 10) and comments regarding contracts issued by reviewers (employee, client, company owner) stored in Reviewers dictionary.

Equipment stores information about hardware (laptops, phones or tablets) that employees use for work. Every products has its specification and condition. When equipment is used by employee, its id appears in Equipment rental entity.

Equipment rental is an entity that is responsible for storying history of rented equipment. It is connected to Employees and Equipment. Stores dates regarding the beginning and end of the rent. However, equipment may be returned on a different day than expected so the actual return date was additionally created.

# 3. Relational schema

## 3.1 Diagram



Data model created in Oracle SQL Developer.

## 3.2 Description of a way of mapping an ER–diagram into a relational schema

In the relational diagram all relations with primary and foreign keys are clearly visible.

Employee’s programming language of choice is stored in a dictionary. One person writes in one language so it is a relation 1:1. The same thing is with a reviewers dictionary, where reviewer can write one opinion about one contract.

Lease contracts, as mentioned before, connects employees with clients. This way it has two foreign keys, but still needs ID as a primary key, because employee might work for client several times. Relations are 1:M, because one employee/client can take part in multiple contracts.

Equipment rental, similarly to lease contracts, need to have ID as a primary key because one piece of equipment can be rented to employee several times. This way two relations are 1:M.

Contract opinions on the other hand, can use their foreign keys as primary keys, as there can only be one opinion written by a single reviewer about specified contract. Because there can be multiple opinions about one contract the relation is M:1.

# 4. DDL script

```
CREATE TABLE clients (
    id          INTEGER NOT NULL,
    name        CLOB NOT NULL,
    tax_number  INTEGER NOT NULL,
    street      CLOB NOT NULL,
    flat_number VARCHAR2(25),
    city        VARCHAR2(25) NOT NULL,
    country     VARCHAR2(25) NOT NULL,
    zip_code    VARCHAR2(6) NOT NULL
);

ALTER TABLE clients ADD CONSTRAINT clients_pk PRIMARY KEY ( id );

CREATE TABLE contract_opinions (
    mark     INTEGER NOT NULL,
    comments CLOB,
    contract INTEGER NOT NULL,
    reviewer INTEGER NOT NULL
);

CREATE UNIQUE INDEX contract_opinions__idx ON
    contract_opinions (
        reviewer
    ASC );

ALTER TABLE contract_opinions ADD CONSTRAINT contract_opinions_pk PRIMARY KEY ( reviewer,

contract );

CREATE TABLE employees (
    id                   INTEGER NOT NULL,
    name                 VARCHAR2(20) NOT NULL,
    surname              VARCHAR2(30) NOT NULL,
    personal_id_number   INTEGER NOT NULL,
    bank_account         CLOB NOT NULL,
    email                VARCHAR2(40) NOT NULL,
    telephone            INTEGER NOT NULL,
    street               CLOB NOT NULL,
    flat_number          VARCHAR2(25),
    city                 VARCHAR2(25) NOT NULL,
    country              VARCHAR2(25) NOT NULL,
    zip_code             VARCHAR2(6) NOT NULL,
    earnings_per_hour    NUMBER NOT NULL,
    programming_language INTEGER NOT NULL,
    date_begin           DATE NOT NULL,
    is_archival          CHAR(1) NOT NULL
);

CREATE UNIQUE INDEX employees__idx ON
    employees (
        programming_language
    ASC );

ALTER TABLE employees ADD CONSTRAINT employees_pk PRIMARY KEY ( id );

CREATE TABLE equipment (
    id            INTEGER NOT NULL,
    name          VARCHAR2(100) NOT NULL,
    specification CLOB,
    condition     CLOB NOT NULL
);

ALTER TABLE equipment ADD CONSTRAINT equipment_pk PRIMARY KEY ( id );

CREATE TABLE equipment_rental (
    id                    INTEGER NOT NULL,
    date_begin            DATE NOT NULL,
    date_end              DATE NOT NULL,
    date_of_actual_return DATE,
    equipment             INTEGER NOT NULL,
    employees             INTEGER NOT NULL
);

ALTER TABLE equipment_rental ADD CONSTRAINT equipment_rental_pk PRIMARY KEY ( id );

CREATE TABLE languages_dictionary (
    id   INTEGER NOT NULL,
    name VARCHAR2(30) NOT NULL
);

ALTER TABLE languages_dictionary ADD CONSTRAINT languages_dictionary_pk PRIMARY KEY
( id );

CREATE TABLE lease_contracts (
    id         INTEGER NOT NULL,
    price      NUMBER NOT NULL,
    date_begin DATE NOT NULL,
    date_end   DATE NOT NULL,
    employee   INTEGER NOT NULL,
    client     INTEGER NOT NULL
);

ALTER TABLE lease_contracts ADD CONSTRAINT lease_contracts_pk PRIMARY KEY ( id );

CREATE TABLE reviewers_dictionary (
    id   INTEGER NOT NULL,
    name VARCHAR2(10) NOT NULL
);

ALTER TABLE reviewers_dictionary ADD CONSTRAINT reviewers_dictionary_pk PRIMARY KEY
( id );

ALTER TABLE contract_opinions
    ADD CONSTRAINT contract_opinions_fk FOREIGN KEY ( contract )
        REFERENCES lease_contracts ( id );

ALTER TABLE lease_contracts
    ADD CONSTRAINT contracts_clients_fk FOREIGN KEY ( client )
        REFERENCES clients ( id );

ALTER TABLE lease_contracts
    ADD CONSTRAINT contracts_employees_fk FOREIGN KEY ( employee )
        REFERENCES employees ( id );

ALTER TABLE employees
    ADD CONSTRAINT employees_languages_fk FOREIGN KEY ( programming_language )
        REFERENCES languages_dictionary ( id );

ALTER TABLE equipment_rental
    ADD CONSTRAINT rental_employees_fk FOREIGN KEY ( employees )
        REFERENCES employees ( id );

ALTER TABLE equipment_rental
    ADD CONSTRAINT rental_equipment_fk FOREIGN KEY ( equipment )
        REFERENCES equipment ( id );

ALTER TABLE contract_opinions
    ADD CONSTRAINT reviewers_opinions_fk FOREIGN KEY ( reviewer )
        REFERENCES reviewers_dictionary ( id );
```