# Project Deliverable 2

*Creating your Database*

*Presented to*

Bettina Kemme

COMP-421 – Database Systems

*By*

Patrick Desmarais (260 329 253)

Simon Hlywa (260 231 002)

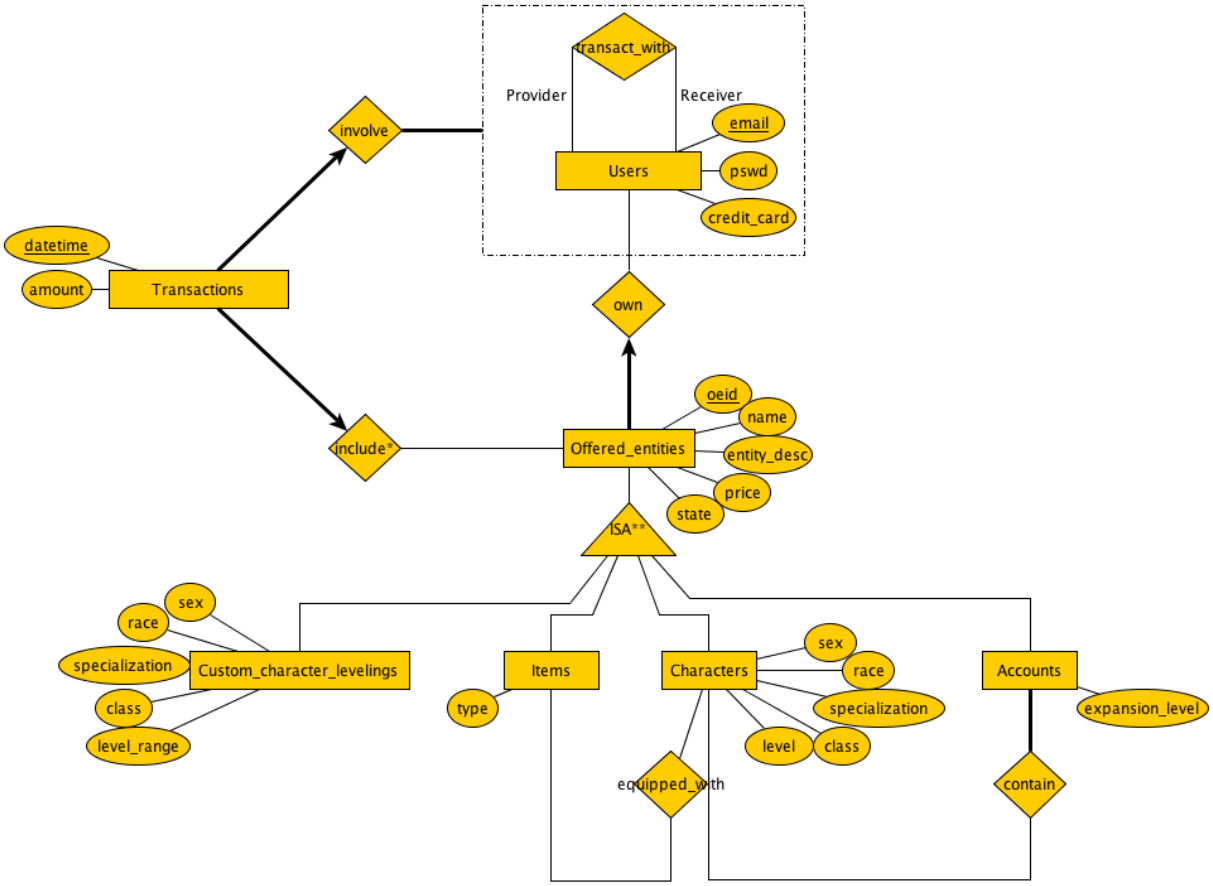
Guillaume Viger (260 309 396)

McGill University

February 15th, 2011

# 1. Corrections

## 1.1. E/R Model



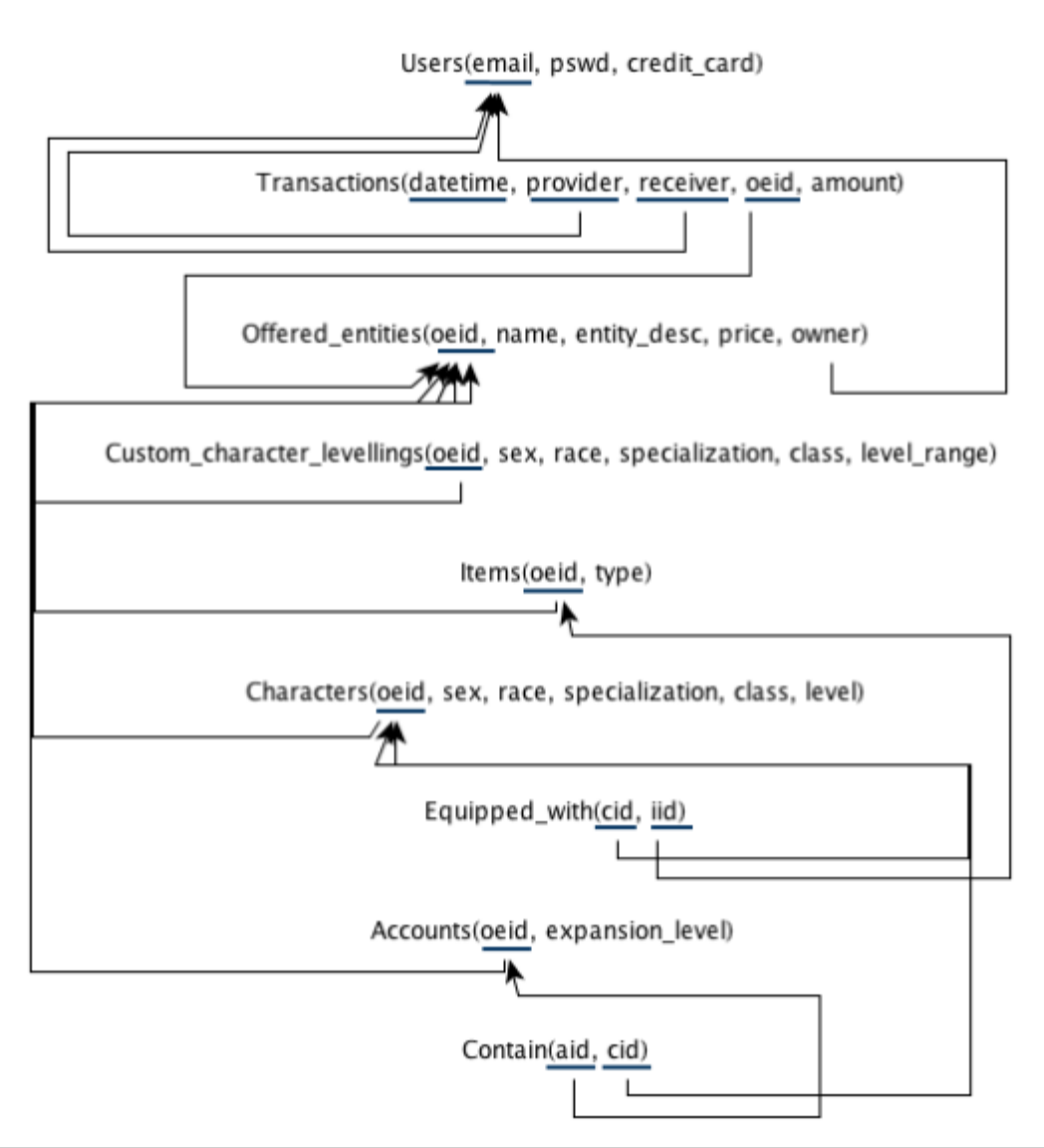

*An offered entity included in a transaction must be owned by the provider of the transaction

**Accounts, Characters, Items and Custom_character_levelings cover Offered_Entities.

Note that since both items and characters can be sold separately from the character or the account, and they can be tied to different characters or accounts; consequently, no participation constraint have been added in relationship sets equipped_with and contain.

The complexity of the E/R model was approved by Prof. Kemme; the number of entity sets and relationship sets was justified.

## 1.2. Relational Schema



# 2. BCNF

Each relational schema has been treated separately to verify if a further decomposition could be executed based on the functional dependencies (FDs) according to the Boyce-Codd Normal Form (BCNF) and the third-normal form (3NF). Paragraphs on possible combinations between tables have also been introduced. Each discussion can be found under the bold components.

**Users**(email, pswd, credit_card)
E : email
P : pswd
C : credit_card

E -> E,P,C
E,P -> E,P,C
*(C -> E,P,C)

$E^+$ = {E,P,C}
$EP^+$ = {E,P,C}
$C^+$ = {E,P,C}

{E} is just a subset of {E,P}; hence, {E,P} is no more a key candidate. Since the primary key {E} is the only key candidate, Users should not be decomposed further.

*Under the assumption that no two users have the same credit card, the third functional dependency suggests that {C} could be a key candidate. However, since Offered_entities were modeled with a user owner and that the owner's email might be displayed along the Offered_entities (for contact purposes), it makes more sense to use {E} as the primary key.

{E} is considered as the only valid key, and {E} -> {E,P,C} respects BCNF by the fact that {E} is a key. It therefore also respects 3NF.

**Transactions**(datetime, provider, receiver, oeid, amount)
D : datetime
P : provider
R : receiver
O : oeid
A : amount

D,P,R,O -> D,P,R,O,A
P,R -> P,R
O,P -> O,P
*(D -> D,P,R,O,A)

$D,P,R,O^{+}$ = {D,P,R,O,A}
$P,R^{+}$ = {P,R}

*Under the assumption that the probability of having two Transactions tuples to be created at the exact same time is very low in terms of the clock accuracy, datetime could be considered as a key candidate. We decided to neglect {D} -> {D,P,R,O,A} due to application semantics.

Transactions could be separated into two tables, as suggested by the aggregation in the E/R model. The additional table would be Transact_with which would contain the relationship between the provider and the receiver. Since {P,R} -> {P,R} is a trivial relationship, the Transact_with table would be redundant.

{D,P,R,O} is considered as the only valid key, and {D,P,R,O} -> {D,P,R,O,A} respects BCNF by the fact that {D,P,R,O} is a primary key. It therefore also respects 3NF. This relation is in BCNF and 3NF.

**Should we combine *Users* and *Transactions*?**
No. Since new Users are not immediately involved in Transactions, we need to keep Users and Transactions table separated. It is not intuitive to create an empty transaction to record a user or to have the Users information be redundantly stored inside all Transactions they are involved with. Finally, the dependency-preservation is satisfied by keeping those two tables separated for any Users insertion.

**Should we have an *Own* table?**
No. Since an offered entity must belong to only one owner (as translated by the key constraint between *Users* and *Offered_entity*), an Own table would be redundant. Only an additional attribute (foreign key to user) in the Offered_entity table is necessary to fulfill the queries described in the functional requirements.

**Offered_entities**(<u>oeid</u>, name, entity_desc, price, owner)
O : oeid
N : name
E : entity_desc
P : price
W : owner
S : state

O -> O,N,E,P,W,S
O, W -> P

$O^+$ = {O,N,E,P,W,S}

Since {N} and {E} do not need to be unique by the functional requirements, only {O} is a key candidate. The latter relation is in BCNF since {O} is a primary key.

**Custom_character_levellings**(<u>oeid</u>, sex, race, specialization, class, level_range)
O : oeid
S : sex
R : race
P : specialization
C : class
L : level_range

O -> O,S,R,P,C,L

$O^+$ = {O,S,R,P,C, L}

Since O is a primary key, it implies all attributes. No other FDs apply in this relation. This relation is in BCNF and in 3NF.

**Items**(oeid, type)
O : oeid
T: type

O -> T

$O^+$ = {O, T}

Since O is a primary key, it implies all attributes. The type T on the other hand does not imply an Offered_entity. This relation is in BCNF and in 3NF since O is the key.

**Characters**(<u>oeid</u>, sex, race, specialization, class, level)
O : oeid
S : sex
R : race
P : specialization
C : class
L : level

O -> O,S,R,P,C,L

$O^+$ = {O,S,R,P,C, L}

O is a primary key, so this relation is in BCNF and in 3NF.

**Equipped_with**(cid, iid)
C: cid
I: iid

I -> I, C

$I^+$={I, C}

All FDs of this relation respect BCNF and 3NF. This relation is in BCNF and 3NF.

**Accounts**(oeid, expansion_level)
O: oeid
E: expansion_level

O -> E

$O^+$={O, E}

All FDs of this relation respect BCNF and 3NF.

**Contain**(aid, cid)
A: aid
C: cid

C -> A

$C^+$={C, A}

All FDs of this relation respect BCNF and 3NF. This relation is in BCNF and 3NF.

# 3. CREATE Statements

Our team will be working on the DB2 database. As such the following CREATE statements follow DB2 syntax. Note that DESCRIBE TABLE outputs follow each CREATE statements for each table.

```
CREATE TABLE Users (
        email VARCHAR(100) NOT NULL PRIMARY KEY,
        pswd VARCHAR (50),
        credit_card CHAR (16)
)
```

```
Column                         Type      Type
name                           schema    name                Length   Scale Nulls
------------------------------ --------- ------------------ -------- ----- ------
EMAIL                          SYSIBM    VARCHAR                100     0 No
PSWD                           SYSIBM    VARCHAR                 50     0 No
CREDIT_CARD                    SYSIBM    CHARACTER               16     0 Yes
```

```
CREATE TABLE Offered_entities (
        oeid BIGINT NOT NULL PRIMARY KEY,
        name VARCHAR(100) NOT NULL,
        entity_desc VARCHAR(1000),
        price REAL,
        state SMALLINT WITH DEFAULT,
        owner VARCHAR(100) NOT NULL,

        FOREIGN KEY (owner) REFERENCES Users
)
```

```
Column                         Type      Type
name                           schema    name                Length   Scale Nulls
------------------------------ --------- ------------------ -------- ----- ------
OEID                           SYSIBM    BIGINT                    8     0 No
NAME                           SYSIBM    VARCHAR                 100     0 No
ENTITY_DESC                    SYSIBM    VARCHAR                1000     0 Yes
PRICE                          SYSIBM    REAL                      4     0 Yes
STATE                          SYSIBM    SMALLINT                  2     0 Yes
OWNER                          SYSIBM    VARCHAR                 100     0 No
```

```
CREATE TABLE Transactions(
        datetime TIMESTAMP NOT NULL WITH DEFAULT,
        provider VARCHAR(30) NOT NULL,
        receiver VARCHAR(30) NOT NULL,
        oeid BIGINT NOT NULL,
        amount REAL NOT NULL,

        PRIMARY KEY (datetime, provider, receiver, oeid),
        FOREIGN KEY (provider) REFERENCES Users ON UPDATE RESTRICT,
        FOREIGN KEY (receiver) REFERENCES Users ON UPDATE RESTRICT,
        FOREIGN KEY (oeid) REFERENCES Offered_entities,
        UNIQUE(oeid)
)
```

```
Column                         Type      Type
name                           schema    name               Length   Scale Nulls
------------------------------  ---------     ------------------ -------- ----- ------
DATETIME                      SYSIBM    TIMESTAMP           10       0 No
PROVIDER                       SYSIBM    VARCHAR               30       0 No
RECEIVER                       SYSIBM    VARCHAR               30       0 No
OEID                                                 SYSIBM    BIGINT                      8
0 No
AMOUNT                         SYSIBM    REAL                        4       0 Yes
```

```
CREATE TABLE  Custom_character_levelings (
        oeid BIGINT NOT NULL PRIMARY KEY,
        sex CHAR(1),
        race VARCHAR(20),
        specialization VARCHAR(50),
        class VARCHAR (50),
        level_range VARCHAR(20),

        FOREIGN KEY (oeid) REFERENCES Offered_entities ON DELETE CASCADE
)
```

```
Column                         Type      Type
name                           schema    name               Length   Scale Nulls
------------------------------ --------- ------------------ -------- ----- ------
OEID                           SYSIBM    BIGINT                    8     0 No
SEX                            SYSIBM    CHARACTER                 1     0 Yes
RACE                           SYSIBM    VARCHAR                  20     0 Yes
SPECIALIZATION                 SYSIBM    VARCHAR                  50     0 Yes
CLASS                          SYSIBM    VARCHAR                  50     0 Yes
LEVEL_RANGE                    SYSIBM    VARCHAR                  20     0 Yes
```

```
CREATE TABLE Items(
        oeid BIGINT NOT NULL PRIMARY KEY,
        type VARCHAR(100),

        FOREIGN KEY (oeid) REFERENCES Offered_entities ON DELETE CASCADE
)
```

| Column name | Type schema | Type name | Length | Scale | Nulls |
|---|---|---|---|---|---|
| OEID | SYSIBM | BIGINT | 8 | 0 | No |
| TYPE | SYSIBM | VARCHAR | 100 | 0 | Yes |

```
CREATE TABLE Characters(
        oeid BIGINT NOT NULL PRIMARY KEY,
        sex CHAR(1), race VARCHAR(20),
        specialization VARCHAR(20),
        class VARCHAR (20),
        level SMALLINT,

        FOREIGN KEY (oeid) REFERENCES Offered_entities ON DELETE CASCADE
)
```

| Column name | Type schema | Type name | Length | Scale | Nulls |
|---|---|---|---|---|---|
| OEID | SYSIBM | BIGINT | 8 | 0 | No |
| SEX | SYSIBM | CHARACTER | 1 | 0 | Yes |
| RACE | SYSIBM | VARCHAR | 20 | 0 | Yes |
| SPECIALIZATION | SYSIBM | VARCHAR | 20 | 0 | Yes |
| CLASS | SYSIBM | VARCHAR | 20 | 0 | Yes |
| LEVEL | SYSIBM | SMALLINT | 2 | 0 | Yes |

```
CREATE TABLE Accounts(
        oeid BIGINT NOT NULL PRIMARY KEY,
        expansion_level SMALLINT,

        FOREIGN KEY (oeid) REFERENCES Offered_entities ON DELETE CASCADE
)
```

| Column name | Type schema | Type name | Length | Scale | Nulls |
|---|---|---|---|---|---|
| OEID | SYSIBM | BIGINT | 8 | 0 | No |
| EXPANSION_LEVEL | SYSIBM | SMALLINT | 2 | 0 | Yes |

```
CREATE TABLE Contain (
        aid BIGINT NOT NULL,
        cid BIGINT NOT NULL,

        PRIMARY KEY(aid, cid)
        FOREIGN KEY (aid) REFERENCES Accounts ON DELETE CASCADE,
        FOREIGN KEY (cid) REFERENCES Characters ON DELETE CASCADE
)
```

| Column name | Type schema | Type name | Length | Scale | Nulls |
|---|---|---|---|---|---|
| AID | SYSIBM | BIGINT | 8 | 0 | No |
| CID | SYSIBM | BIGINT | 8 | 0 | No |

```
CREATE TABLE Equipped_with (
        cid BIGINT NOT NULL,
        iid BIGINT NOT NULL,

        PRIMARY KEY(cid, iid)
        FOREIGN KEY (cid) REFERENCES Characters ON DELETE CASCADE,
        FOREIGN KEY (iid) REFERENCES Items ON DELETE CASCADE
)
```

| Column name | Type schema | Type name | Length | Scale | Nulls |
|---|---|---|---|---|---|
| CID | SYSIBM | BIGINT | 8 | 0 | No |
| IID | SYSIBM | BIGINT | 8 | 0 | No |

## Constraints that couldn’t be expressed...

Accounts, Characters, Items and Custom_character_levelings cover Offered_Entities; however, there is no guarantee that an offered entity is created without being specified as an account, a character, an item or a custom character leveling.

# 4. INSERT Statements

We will insert five Characters into the table. To do so we will insert 5 Users and 5 Offered_entities. The select statement will only be on the 5 Characters added.

```
INSERT INTO Users \
VALUES \
    ('lonelygirl15@example.com', 'password', '123456'), \
    ('merlin428@example.com', 'hello12', '54321'), \
    ('sherlock_9382@aol.com', 'wow12','0765675'), \
    ('tinyT1m@example255.com', 'pword32', '74672378'), \
    ('tomB0mbadil2345@lotr.com', '2344','3423234234')
DB20000I  The SQL command completed successfully.



INSERT INTO Offered_entities \
VALUES \
    (9100, 'Borag of the Many', 'Hailing from Gilneas, this Worgen character is ready to inflict
pain on your opponents.Buy him!', \
    50.00, 0, 'lonelygirl15@example.com'), \
    (9101, 'Melfor the Sneaky', 'Silent Death. This character is ideal in combination with a DPS
Mage gnome and a Warrior Tank', \
    59.99, 0, 'merlin428@example.com'), \
        (9102, 'Balthazar07', 'This character has completed all the Azure Myst quests.', \
    39.99, 0, 'sherlock_9382@aol.com'), \
    (9103, 'Bobo_the_fearsome', 'Buy this character it is fearsome! Not really, but his equipment
is pretty good', \
    25.50,1, 'tinyT1m@example255.com'), \

(9104, 'Galadriel of Rivendell', 'Completes the collection of Lord of the Ring characters',75.99,
1, 'tomB0mbadil2345@lotr.com')
DB20000I  The SQL command completed successfully.



INSERT INTO Characters \
VALUES    ('9100', 'Male', 'Worgen', 'DPS', 'Warrior', '70'), \
         ('9101', 'Female', 'Night Elf', 'DPS', 'Warrior', '85'), \
          ('9102', 'Male', 'Draenei', 'Tank', 'Paladin', '20'), \
          ('9103', 'Male', 'Gnome', 'DPS', 'Warrior', '40'), \
         ('9104', 'Female', 'Night Elf', 'Healer', 'Priest', '85')
DB20000I  The SQL command completed successfully.
```

Print-outs of the select statement:

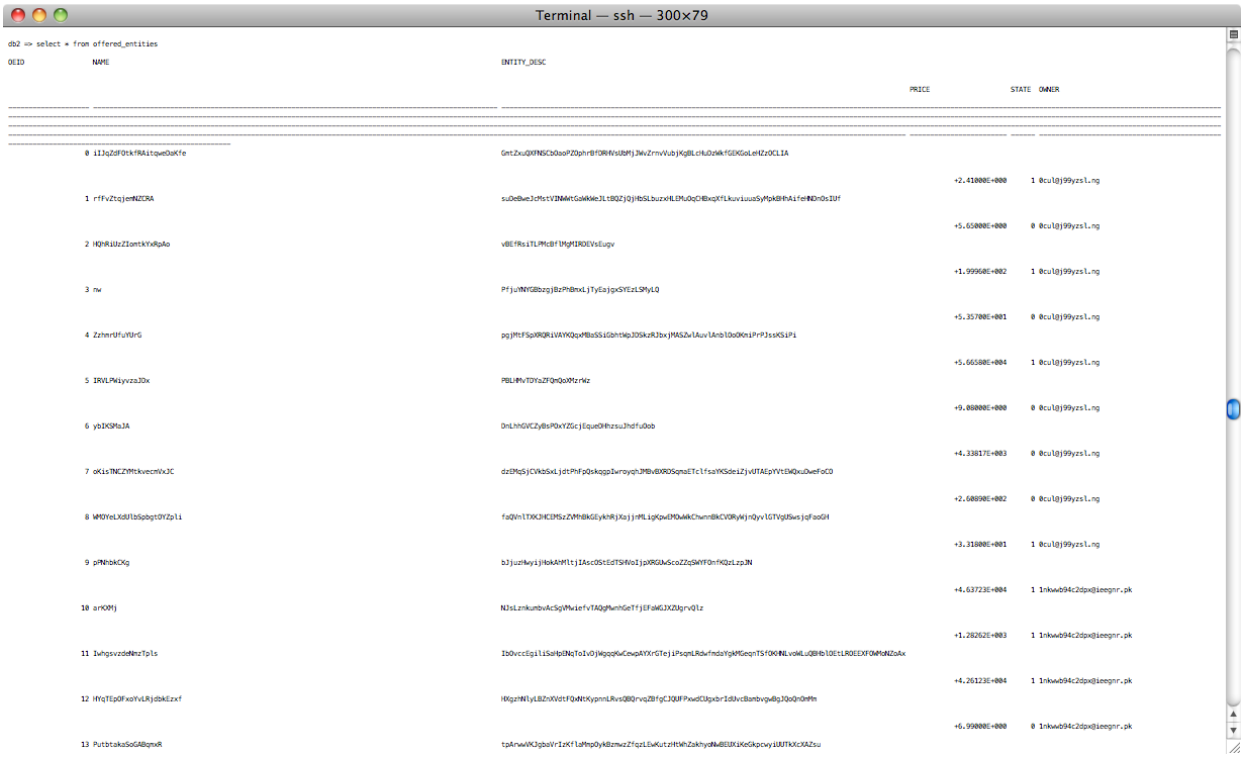
```
db2 => select * from Characters

OEID                 SEX RACE                 SPECIALIZATION       CLASS                LEVEL
-------------------- --- -------------------- -------------------- -------------------- ------
                9100 M   Worgen               DPS                  Warrior                  70
                9101 F   Night Elf            DPS                  Warrior                  85
                9102 M   Draenei              Tank                 Paladin                  20
                9103 M   Gnome                DPS                  Warrior                  40
                9104 F   Night Elf            Healer               Priest                   85

  5 record(s) selected.
```

# 5. Substantial INSERT Statements

The code to generate the data can be found in DataGenerator.java .

The 9000-tuple relations were chosen for the Offered_entities and Characters tables. Here is a sample of the generated data for those relations:

- Sample for generated tuples in Offered_entities




- Sample for generated tuples in Characters

```
db2 => select * from characters

OEID                 SEX RACE                 SPECIALIZATION       CLASS                LEVEL
-------------------- --- -------------------- -------------------- -------------------- ------
                   0 M   Undead               Subtlety             Warlock                  52
                   1 M   Draenei              Fire                 Warrior                  82
                   2 M   Undead               Shadow               Paladin                  18
                   3 M   Orc                  Subtlety             Druid                    20
                   4 F   Draenei              Subtlety             Hunter                   49
                   5 M   Blood Elf            Subtlety             Rogue                    83
                   6 M   Human                Shadow               Hunter                   81
                   7 M   Worgen               Arcane               Hunter                   29
                   8 M   Draenei              Holy                 Rogue                    25
                   9 M   Troll                Resto                Hunter                   58
...
```

The 1000-tuple relation was chosen to be Users. Here is a sample of the records generated and the script showing the load of data:

```
db2 => import from users.del of del insert into cs421g14.users
SQL3109N  The utility is beginning to load data from file "users.del".

SQL3110N  The utility has completed processing.  "1000" rows were read from
the input file.

SQL3221W  ...Begin COMMIT WORK. Input Record Count = "1000".

SQL3222W  ...COMMIT of any database changes was successful.

SQL3149N  "1000" rows were processed from the input file.  "1000" rows were
successfully inserted into the table.  "0" rows were rejected.


Number of rows read         = 1000
Number of rows skipped      = 0
Number of rows inserted     = 1000
Number of rows updated      = 0
Number of rows rejected     = 0
Number of rows committed    = 1000

db2 => select * from users

EMAIL
PSWD                                                CREDIT_CARD
--------------------------------------------------------------------------------------------------
--- -------------------------------------------------- ----------------
0selgqmkxoicktt@p8v.vs
5dY2N3IGT75aq7AjO                                   4723604617152524
1uts@olmhpjgj.bsn
OTO9aZydX3WZWtkuzTwk5                               0070384544636201
2nile8hm3nfiz@tqi8ejixmg.nk
NB6cD7q8YYOjYnueinr6WR                              2033080864494555
3g8tagssrrr9faialsu3dsm@y6tdclvoz6.pr
4DQYGne4                                            1015101447750609
4nodx8lczy@ilvpaho.iqk
aMCbV3mznlo                                         5838020113736677
5zxmk8m6alo3t2o4g@xhgobksqp.abo
zpchnCyAuvmlcavA                                    1105260659319378
6j3af94gdlngs@9egfi.odb
111ZVHC6QQ9bNF                                      2024741734553961
7dclp9eursgrblqrotsaon@mvxd.gk
khClpOa9uhV                                         9089232437196157
8v2yndx3@amnpkkh.htk
hQZ253qyjexyVu                                      2989630424823333
95eujpfzke4zopa5h@lwyhsce.zmz
TZiVPOQMPnaf6Abuv                                   7634833512523369
10mxcozcz9g@2dl.pk
ZrIqyprMZEspecj                                     5889421597565483
…

  1000 record(s) selected.
```

Unfortunately, the table columns do not make it be readable while respecting this document's margins.