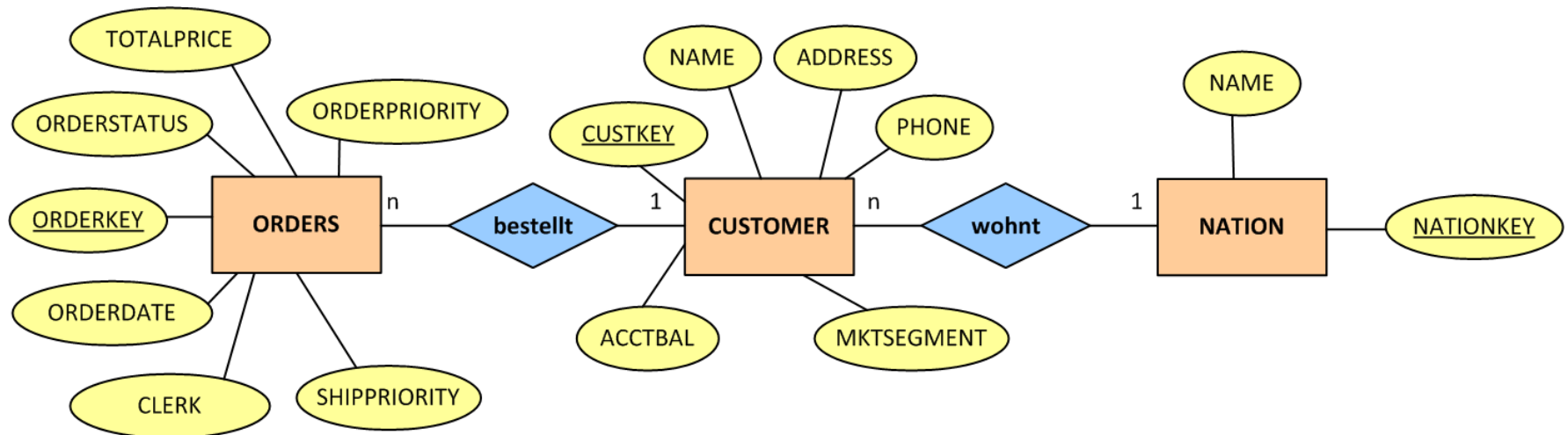


Aufgabe 12.1: JDBC - Datenbankzugriff in Java

Beispielquery

Schema:

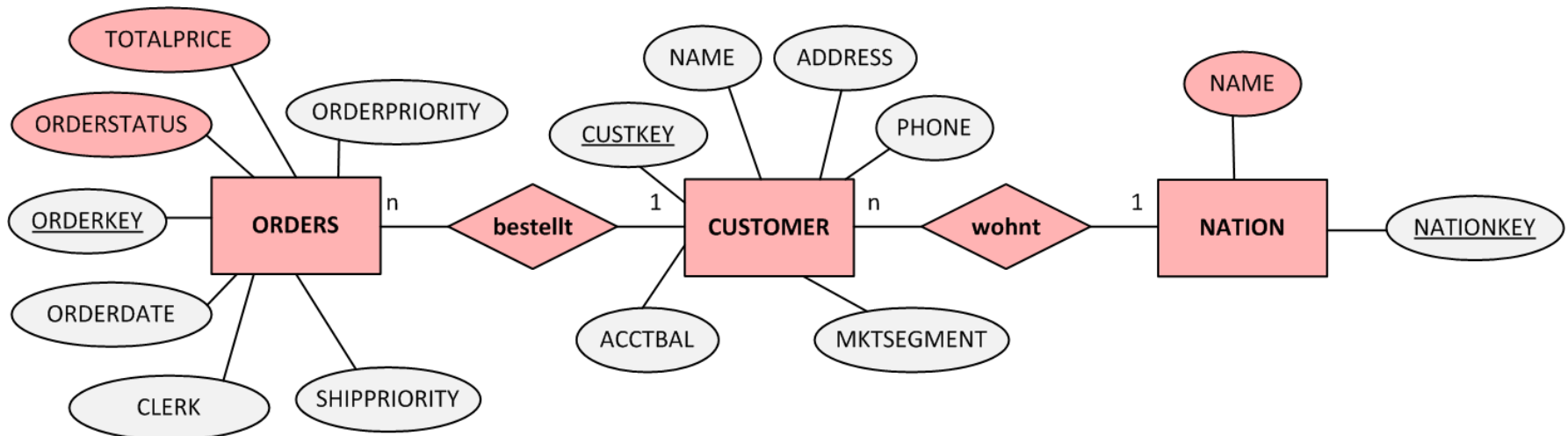


- „Wie ist der Name des Landes, für welches der Wert der nicht ausgelieferter Bestellungen maximal ist?“

Aufgabe 12.1: JDBC - Datenbankzugriff in Java

Beispielquery

- Für diese Query relevante Bestandteile des Schemas:



- „Wie ist der Name des Landes, für welches der Wert der nicht ausgelieferter Bestellungen maximal ist?“

Aufgabe 12.1: JDBC - Datenbankzugriff in Java

„Wie ist der Name des Landes, für welches der Wert der nicht ausgelieferter Bestellungen maximal ist?“

■ Selektion:

1. Nicht ausgelieferte Bestellungen
2. `orderstatus = '0'`
3. Maximaler Wert der Bestellungen pro Land
4. 1. Möglichkeit: Nested Query zur Ermittlung des Maximums
5. 2. Möglichkeit: Ergebnis absteigend sortieren, erstes Resultat auswählen

■ Projektion:

- `name` des Landes: `nation.name`

■ Join:

- Bedingung:
`o.custkey = c.custkey`
`AND c.nationkey = n.nationkey`

Aufgabe 12.1: JDBC - Datenbankzugriff in Java

1. Möglichkeit: Nested Query zur Ermittlung des Maximums

```
SELECT n.name, SUM(o.totalprice) AS total
FROM orders o, customer c, nation n
WHERE o.custkey = c.custkey AND c.nationkey = n.nationkey
      AND o.orderstatus = '0'
GROUP BY n.name
HAVING total =
(
  SELECT max(total)
  FROM
  (
    SELECT n.name, SUM(o.totalprice) AS total
    FROM orders o, customer c, nation n
    WHERE o.custkey = c.custkey AND c.nationkey = n.nationkey
          AND o.orderstatus = '0'
    GROUP BY n.name
  ) allnations
)
```

Diese Lösung ist kompliziert!

Aufgabe 12.1: JDBC - Datenbankzugriff in Java

2. Möglichkeit: Ergebnisse absteigend sortieren

```
SELECT n.name, SUM(o.totalprice) AS total
FROM orders o, customer c, nation n
WHERE o.custkey = c.custkey AND c.nationkey = n.nationkey
      AND o.orderstatus = 'O'
GROUP BY n.name
ORDER BY total DESC
LIMIT 1
```

Einfachere Lösung

Aufgabe 12.1: JDBC - Datenbankzugriff in Java

Abweichende Semantik

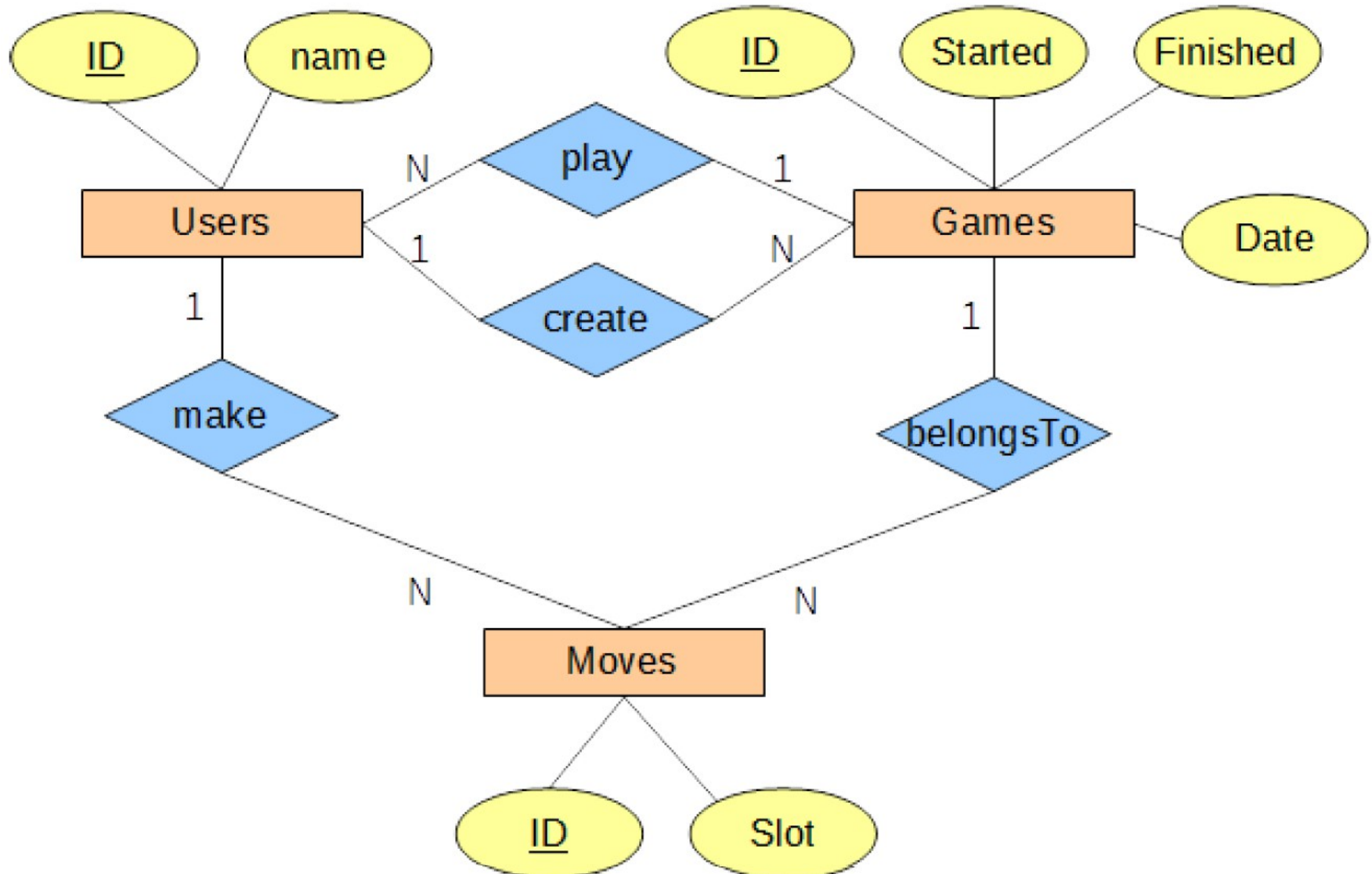
- 1. Möglichkeit: Alle Länder, für die der Wert der offenen Bestellungen maximal ist. Es kann mehrere Ergebnisse geben, falls die Summe bei mehreren Ländern gleich ist!
- 2. Möglichkeit: Es wird nur eines der Länder ausgegeben, bei dem der Wert maximal ist. Falls es mehrere solcher Länder gibt, ist nicht definiert welches ausgegeben wird!

Aufgabe 12.1: JDBC - Datenbankzugriff in Java

Fazit

- Die Semantik der Queries ist etwas anders, aber beides sind Lösungen für die gestellte Aufgabe
- Oft gibt es mehrere Lösungen in SQL für dasselbe Problem
- Lösungen unterscheiden sich oft stark in der Komplexität

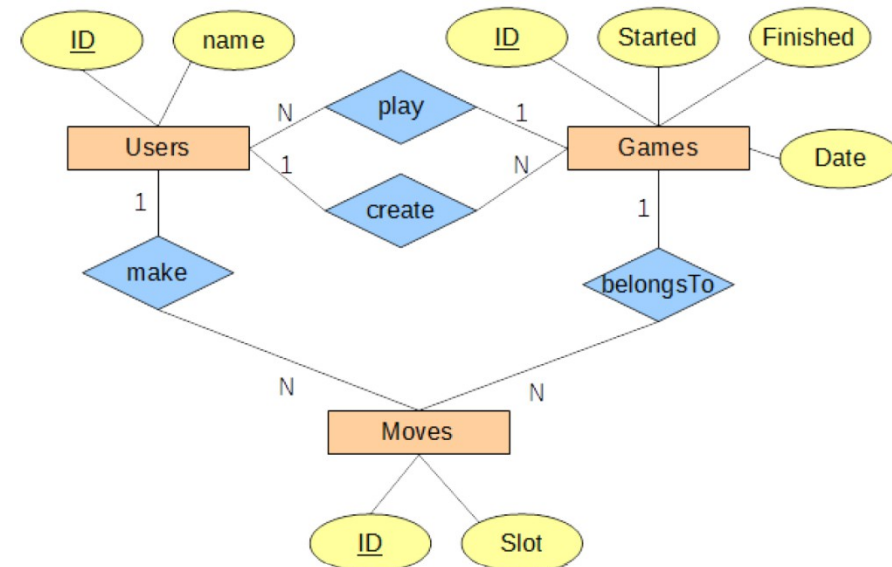
Aufgabe 12.2: Vier Gewinnt (Schema)



Aufgabe 12.2 (a): Relationen

Entities

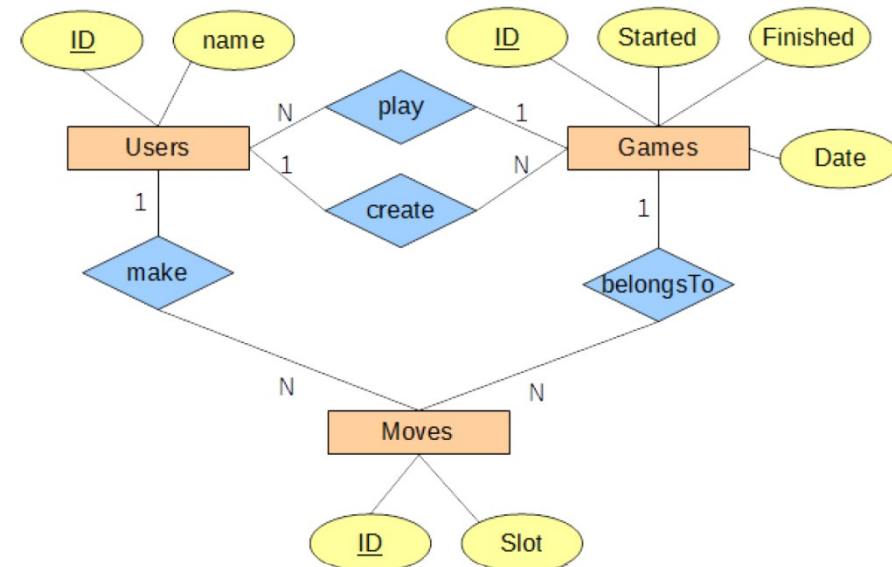
- **Users** (userID, name)
- **Games** (gameID, started, finished, date)
- **Moves** (moveID, slot)



Aufgabe 12.2 (a): Relationen

Relationships

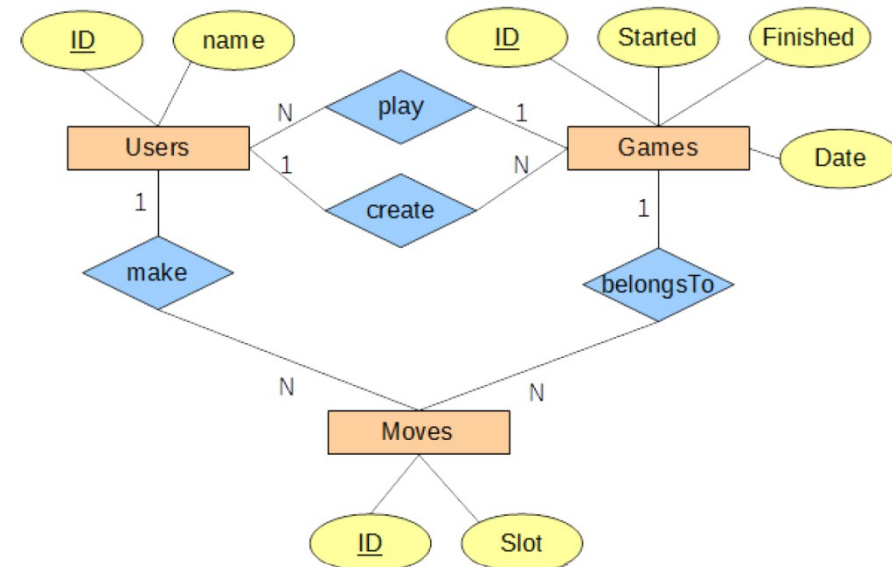
- **play** (gameID, userID)
- **create** (gameID, createdByUserID)
- **make** (moveID, userID)
- **belongsTo** (moveID, gameID)



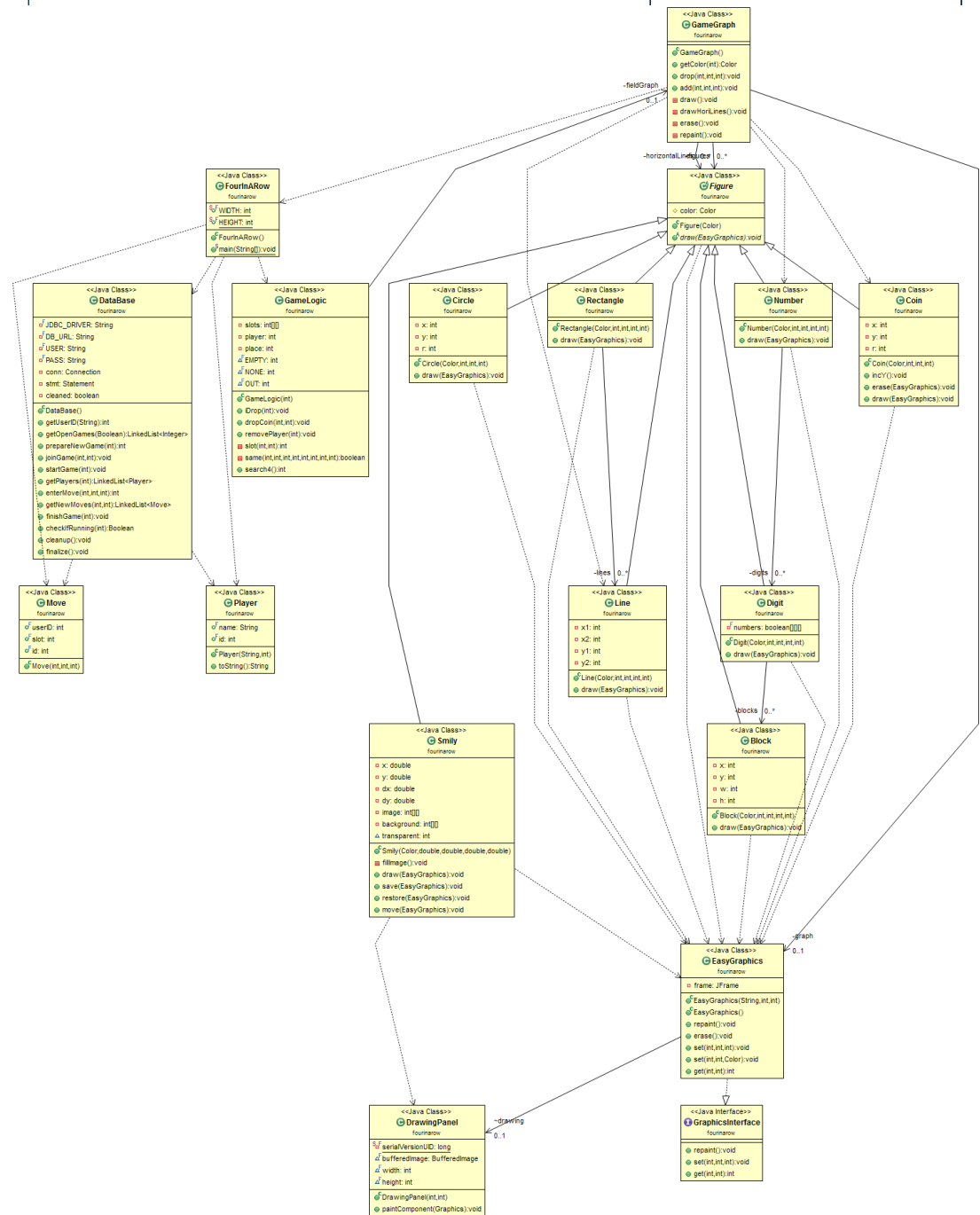
Aufgabe 12.2 (a): Relationen

Kombiniere Relationen mit denselben Schlüsseln

- **Users** (userID, name, playGameID)
- **Games** (gameID, started, finished, date, createdByUserID)
- **Moves** (moveID, slot, gameID, userID)

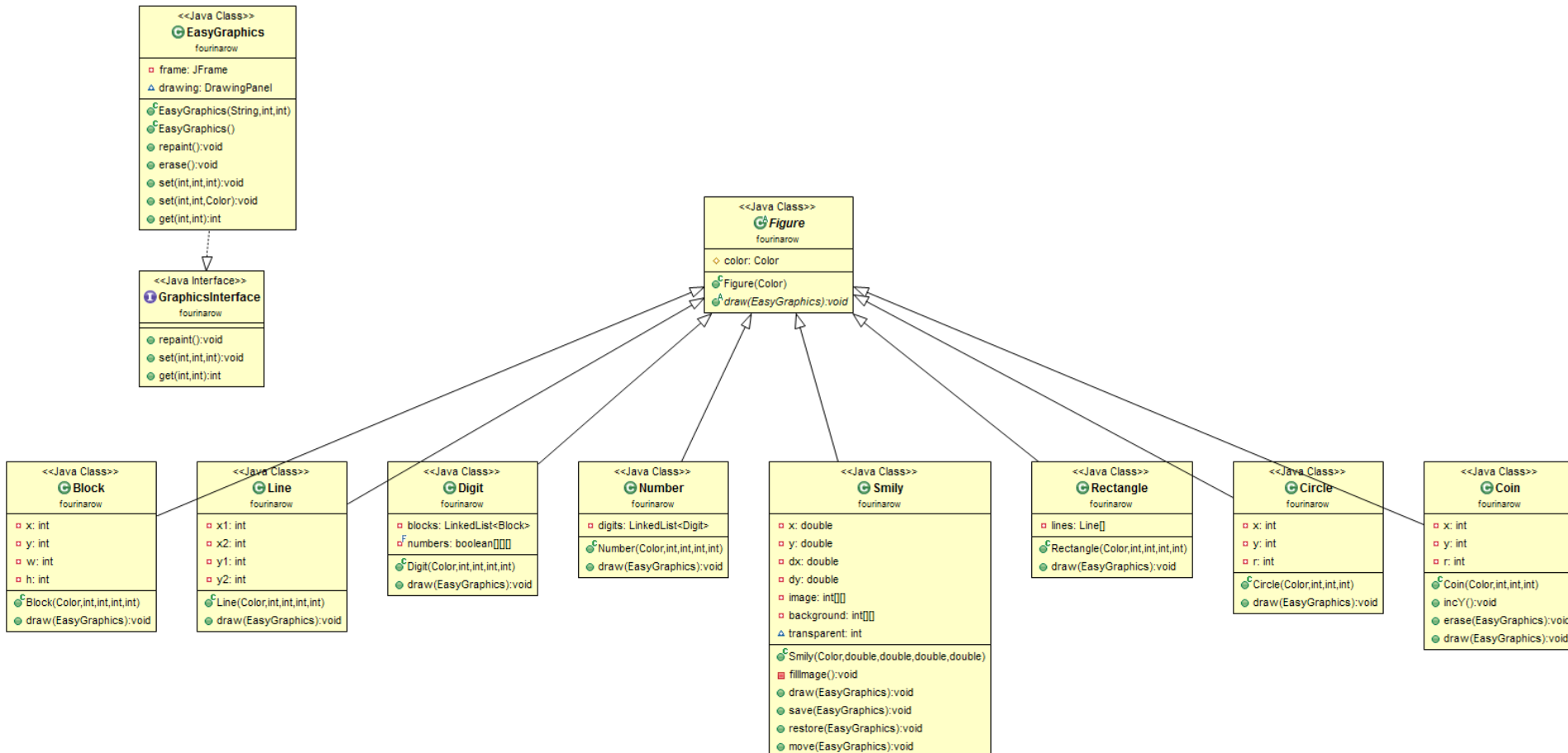


Aufgabe 12.2 (c): Klassendiagramm



Tool: ObjectAid UML Explorer

Aufgabe 12.2 (c): Klassendiagramm (Vererbung)



Aufgabe 12.2 (d): Queries

getUserID()

Query: Zeilen der Tabelle Users mit Name=<name>

```
"SELECT * FROM Users  
WHERE Name = \"\"+name+\"\""
```

Einfügen: neue Zeile (Name, GameID) in Tabelle Users mit Werten (<name>, <gameId>)

```
"INSERT INTO Users (Name, GameID)  
VALUES  
(\"\"+name+\"\", \"\"+gameID+\"\")"
```

Aufgabe 12.2 (d): Queries

Query: Zeilen der Tabelle Games mit ID = <gameId>

```
"SELECT * FROM Games  
WHERE ID = \"\"+gameID+\"\""
```

getOpenGames()

**Query: Spalte ID der Zeilen der Tabelle Games mit
Started = "0"**

```
"SELECT ID FROM Games WHERE Started = \"0\""
```

Aufgabe 12.2 (d): Queries

prepareNewGame()

Einfügen: neue Zeile (Started, Finished, Creator) in Tabelle Games mit Werten ("0", "0", <userId>)

```
"INSERT INTO Games (Started, Finished,  
Creator)  
VALUES (\\"0\\", \\"0\\", \\""+userID+"\\")"
```

Query: Spalte ID der Zeilen der Tabelle Games mit Started = "0", Finished = "0" und Creator = <userId>, nach ID geordnet

```
"SELECT ID FROM Games WHERE Started = \\"0\\"  
AND Finished = \\"0\\"  
AND Creator = \\""+userID+"\\" ORDER BY ID"
```


Aufgabe 12.2 (d): Queries

joinGame()

Update: Spalte GameID neuer Wert <gameID> in Tabelle Users der Zeilen mit ID = <userID>

```
"UPDATE Users SET GameID = "+gameID+"  
WHERE ID = "+userID
```

startGame()

Update: Spalte Started neuer Wert "1" in Tabelle Games der Zeilen mit ID = <gameID>

```
"UPDATE Games SET Started = \"1\"  
WHERE ID = "+gameID
```

Aufgabe 12.2 (d): Queries

getPlayers()

Query: Spalten ID und Name der Zeilen von Tabelle Users mit Gameld = <gameID>, geordnet nach ID

```
"SELECT ID, Name FROM Users  
WHERE GameID = \"\"+gameID+\"\" ORDER BY ID"
```

enterMove()

Einfügen: (UserID, Slot, Gameld) in Tabelle Moves mit Werten (<userID>, <slot>, <gameID>)

```
"INSERT INTO Moves (UserID, Slot, GameID)  
VALUES  
(\"\"+userID+\"\", \"\"+slot+\"\", \"\"+gameID+\"\"  
)"
```

Aufgabe 12.2 (d): Queries

Query: Spalte ID in Tabelle Moves aller Spalten mit UserID = <userID>, Slot = <slot> und GameID = <gameID>, geordnet nach ID

```
"SELECT ID FROM Moves WHERE UserID  
= \"\"+userID+\"\" AND Slot = \"\"+slot+\"\" AND  
GameID = \"\"+gameID+\"\" ORDER BY ID"
```

getNewMoves()

Query: Spalten UserID, Slot, ID in Tabelle Moves der Zeilen mit GameID = <gameID> und ID = <lastID>, geordnet nach ID

```
"SELECT UserID, Slot, ID FROM Moves  
WHERE GameID = \"\"+gameID+\"\" AND  
ID > \"\"+lastID+\"\" ORDER BY ID"
```

Aufgabe 12.2 (d): Queries

finishGame()

**Update: Spalte Finished mit Wert "1" in Tabelle Games
der Zeilen mit ID = <gameID>**

```
"UPDATE Games SET Finished = \"1\"  
WHERE ID = \"\"+gameID+\"\" "
```

checkIfRunning()

**Query Spalten Started, Finished der Tabelle Games,
Zeilen mit ID = <gameID>**

```
"SELECT Started, Finished FROM Games  
WHERE ID = \"\"+gameID+\"\" "
```