MTAT.03.105
Introduction to Databases

Lecture #6
Constraints

Ljubov Jaanuska
(ljubov.jaanuska@ut.ee)
Lecture 5. Summary

- E-R model according to Crow’s Foot notation
- Model normalization
Lecture 2-3. What do we know about SQL?

- Create a database
- Create a table
  - Column name
  - Data type
    - char, varchar, long varchar
    - tinyint, smallint, integer, begint, decimal, numeric, float, real, double
    - date, time, datetime
    - bit, binary, long binary
  - [default values – autoincrement, timestamp, current timestamp]
  - [constraints – unique, not null, check, **primary key, foreign key**]
- INSERT, SELECT, UPDATE, DELETE rows
- Alter table add constraint/column
Check

- Primary keys
- No columns called name or id
- Data in tables
- Table Club: unique names

- Table Player: unique combinations of names and surnames
- Table Game: game results for both players (white and black chess pieces) must be 0, 1 or 2
- Table Game: sum of both player results must be equal to 2
- Table Tournament: unique names
Revision quiz

https://kahoot.it
Lecture 6. What will you learn

- Add foreign keys
- Learn PCA database (its tables and attributes)
Problem statement

• **Primary key** - is a field in a table which uniquely identifies each row/record in a database table
  • must contain unique values
  • column cannot have NULL values
  • one primary key per table, which may consist of single or multiple fields

• **Foreign key** - is a column (or columns) that references a column with a primary key/unique constraint of another table
  • only values that appear in the reference column are permitted (referential integrity of the data)
Primary key

• Primary key
  
  ALTER TABLE Player
  ADD CONSTRAINT pk_player
  PRIMARY KEY (id);

• Foreign key
  
  ALTER TABLE Player
  ADD CONSTRAINT fk_player_club
  FOREIGN KEY (club_id)
  REFERENCES Club(id)
  ON DELETE CASCADE ON UPDATE CASCADE
ON DELETE ... ON UPDATE

How does DBMS react if the referenced rows in the suptable are changed?

• **RESTRICT** - generates an error and prevents the modification if an attempt to alter a referenced value occurs (*no action*)
• **CASCADE** - the child data is also deleted/updated when the parent data are deleted/updated
• **SET NULL** - the child data is set to NULL when the parent data are deleted or updated
• **SET DEFAULT** - the child data is set to default values (as specified in the table definition) when the parent data are deleted or updated
ON DELETE ... ON UPDATE

```sql
ALTER TABLE <table_name>
ADD CONSTRAINT <fk_name>
FOREIGN KEY({column_name})
REFERENCES <parent_table>[(<parent_table_column>)]
[ON DELETE {action}][ON UPDATE {action}];
```

**ACTION:**
- RESTRICT
- CASCADE
- SET NULL
- SET DEFAULT

<table>
<thead>
<tr>
<th></th>
<th>Arvo</th>
<th>Mets</th>
<th>(NULL)</th>
<th>51</th>
</tr>
</thead>
<tbody>
<tr>
<td>72</td>
<td>Maari</td>
<td>Mustikas</td>
<td>(NULL)</td>
<td>54</td>
</tr>
<tr>
<td>73</td>
<td>Piotr</td>
<td>Pustota</td>
<td>(NULL)</td>
<td>59</td>
</tr>
<tr>
<td>74</td>
<td>Kalle</td>
<td>Kivine</td>
<td>(NULL)</td>
<td>57</td>
</tr>
<tr>
<td>75</td>
<td>Malle</td>
<td>Maasikas</td>
<td>(NULL)</td>
<td>57</td>
</tr>
<tr>
<td>76</td>
<td>Linda</td>
<td>Sammal</td>
<td>(NULL)</td>
<td>58</td>
</tr>
<tr>
<td>77</td>
<td>Arvo</td>
<td>Angervaks</td>
<td>(NULL)</td>
<td>59</td>
</tr>
<tr>
<td>78</td>
<td>Andrei</td>
<td>Sosnov</td>
<td>(NULL)</td>
<td>59</td>
</tr>
<tr>
<td>79</td>
<td>Helina</td>
<td>Hiis</td>
<td>(NULL)</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>Laudnikud</td>
<td>Tartu</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>Ajurebend</td>
<td>Tartu</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>Ruudu Liine</td>
<td>Tartu</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>Vöitmatu Valge</td>
<td>Tartu</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>58</td>
<td>Valge Mask</td>
<td>Tartu</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>Musta kivi kummardajad</td>
<td>Tartu</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
Task

Add 4 foreign keys (pay attention to actions)

- Player -> Club
  (row deletion is not allowed, updates are made in both tables)

- Game -> Tournament
  (if the tournament is deleted, the corresponding games are also removed; updates are made in both tables)

- Game -> Player (for the ones who play with white chess pieces)
  (row deletion is not allowed, updates are made in both tables)

- Game -> Player (for the ones who play with black chess pieces)
Task

- Try to delete club ‘Laudnikud’
- Try to update club with id=57 to id=60
- Change data type (varchar(50)) of column city on table Club to varchar(70)
- Create table District (d_id integer, d_name varchar(100))
  - d_id has a primary key and autoincrement values
  - d_name must be unique
  - on data insertion both values must be specified
Task

• Add all cities/venues from table Club and Tournament into d_name of table District:

\[
\text{INSERT INTO District (d_name) SELECT city FROM Club UNION SELECT venue FROM Tournament}
\]

• Add into table Club column district_id
• Update values in column district_id of table Club by the corresponding d_id values from table District:

\[
\text{UPDATE Club SET district_id = (SELECT d_id FROM District WHERE District.d_name= Club.city)}
\]
Task

• Add a foreign key to table Club referencing to table District
• Add into table Tournament column district_id
• Update values in column district_id of table Tournament by the corresponding d_id values from table District:

```
UPDATE Tournament SET district_id = (SELECT d_id FROM District WHERE District.d_name = Tournament.venue)
```

• Add a foreign key to table Tournament referencing to table District
• Add into your database club ‘SQL club’ which is located in Tartu.
• Add yourself into the database as a player of club ‘SQL club’.
Task

- Please submit your `.db` file and `.log` file by next session via Moodle