Order man System:

Datenbank:

Anfangs wird alles von einem Techniker per SQL konfiguriert. Also die DB soll alles enthalten, was auf dem Event angeboten wird. Auch soll gespeichert sein, wie viel von allem übrig ist.

Der Techniker soll immer per Knopfdruck nachschauen können, wie viel von allem noch übrig ist.

Zusätzlich soll alles was rausgeht, also was verkauft wird, mitgeloggt werden. Dies soll mit einem Trigger realisiert werden. Bei jedem senden soll der letzte Tupel ausgedruckt werden. Die Daten die gespeichert werden sollen zur Auswertung helfen, die zum Schluss des Events stattfindet.

Es braucht insgesamt 6 Tables: Tisch, Getränke, Essen, Logger, Kellner, Drucker

Tables:

Tisch:

|  |  |
| --- | --- |
| tisch\_nr | rest\_betrag |
| 1 | 20,00 |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |

Kellner

|  |  |
| --- | --- |
| kellner\_id | name |
| 1 | Hans |
| 2 | Josef |

Getränke:

|  |  |  |
| --- | --- | --- |
| getraenk\_id | typ | preis |
| 1 | cola | 2,00 |
| 2 | bier | 3,00 |
|  |  |  |

Essen:

|  |  |  |
| --- | --- | --- |
| Essen\_id | Typ | preis |
| 1 | nudl | 8,00 |
| 2 | knedl | 10,00 |
|  |  |  |

Logger:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id | kellner\_id | tisch\_nr | essen | getraenk | zusätzlich |
| 1 | 4 | 3 | null | bier | null |
| 2 | 2 | 5 | knedl | cola | null |

Jede einzelne Bestellung wird in einem eigenen Tupel geschrieben. Tupel mit selber tisch\_id werden zusammengefasst und in einer neuen Datenbank geschrieben, die zum Drucken bereitgestellt wird.

Drucker:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| tisch\_id | Kellner\_name | essen | getraenk | zusätlich |
|  |  |  |  |  |
|  |  |  |  |  |

CREATE TABLE tisch (t\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY);

CREATE TABLE kellner (k\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY, name VARCHAR(50), password VARCHAR(100));

CREATE TABLE bestellung (b\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY, tisch INT, kellner INT, tisch\_id INT FOREIGN KEY REFERENCES tisch(t\_id) ), kellner\_id INT FOREIGN KEY REFERNCES kellner(k\_id));

CREATE TABLE artikel (a\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY, name VARCHAR(50), preis DOUBLE, sparten\_id INT FOREIGN KEY REFERNCES sparte(s\_id));

CREATE TABLE sparte (s\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY, name VARCHAR(50));

CREATE TABLE drucker (d\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY, name VARCHAR(50), standort VARCHAR(50));

CREATE TABLE drucker\_artikel (da\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY, dr\_id INT FOREIGN KEY REFERENCES drucker(d\_id), art\_id INT FOREIGN KEY REFERENCES artikel(a\_id));

CREATE TABLE artikel\_bestellung (arbe\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY, art\_id INT FOREIGN KEY REFERENCES artikel(a\_id), bestellung\_id INT FOREIGN KEY REFERENCES bestellung(b\_id), bezahlt BOOLEAN, zusatzlich VARCHAR(300));