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

Bei Sparte wird noch eine Bool Variable hinzugefügt, welche zurückgibt ob es eine Speise oder ein Getränk ist.

Tables:

Tisch:

|  |
| --- |
| t\_id |
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |

Kellner

|  |  |  |  |
| --- | --- | --- | --- |
| k\_id | name | password | anz\_bestellungen |
| 1 | Hans | hans123 | 23 |
| 2 | Josef | josef123 | 34 |

Bestellung:

|  |  |  |
| --- | --- | --- |
| b\_id | tisch\_id | kellner\_id |
|  |  |  |
|  |  |  |

Artikel:

|  |  |  |  |
| --- | --- | --- | --- |
| a\_id | name | sparten\_id | verkauft |
|  |  |  |  |
|  |  |  |  |

Sparte:

|  |  |  |
| --- | --- | --- |
| s\_id | name | speise |
|  |  |  |
|  |  |  |

Drucker:

|  |  |
| --- | --- |
| d\_id | name |
|  |  |
|  |  |

Drucker\_artikel:

|  |  |  |
| --- | --- | --- |
| da\_id | dr\_id | art\_id |
|  |  |  |
|  |  |  |

Artikel\_bestellung:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| arbe\_id | art\_id | bestellung\_id | bezahlt | zusatzlich |
|  |  |  |  |  |
|  |  |  |  |  |

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.

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

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

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

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

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

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(dr\_id) REFERENCES drucker(d\_id), art\_id INT, FOREIGN KEY(art\_id) REFERENCES artikel(a\_id));

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

INSERT INTO tisch (t\_id) VALUES (1);

INSERT INTO tisch (t\_id) VALUES (2);

INSERT INTO tisch (t\_id) VALUES (3);

INSERT INTO kellner (name, password, anz\_bestellungen) VALUES ('Hans', 'hans123', 23);

INSERT INTO kellner (name, password, anz\_bestellungen) VALUES ('Josef', 'josef123', 34);

INSERT INTO bestellung (tisch\_id, kellner\_id) VALUES (2, 1);

INSERT INTO bestellung (tisch\_id, kellner\_id) VALUES (3, 2);

INSERT INTO artikel (name, sparten\_id, verkauft) VALUES ('Cola', 1, 200);

INSERT INTO artikel (name, sparten\_id, verkauft) VALUES ('Saft', 1, 150);

INSERT INTO artikel (name, sparten\_id, verkauft) VALUES ('Mineral', 1, 300);

INSERT INTO artikel (name, sparten\_id, verkauft) VALUES ('Fanta', 1, 100);

INSERT INTO artikel (name, sparten\_id, verkauft) VALUES ('Sprite', 1, 225);

INSERT INTO artikel (name, sparten\_id, verkauft) VALUES ('Bier', 2, 500);

INSERT INTO artikel (name, sparten\_id, verkauft) VALUES ('Kerner', 2, 120);

INSERT INTO artikel (name, sparten\_id, verkauft) VALUES ('Sylvaner', 2, 130);

INSERT INTO artikel (name, sparten\_id, verkauft) VALUES ('St. Magdalener', 2, 160);

INSERT INTO sparte (name, speise) VALUES ('analkoholisches', false);

INSERT INTO sparte (name, speise) VALUES ('alkoholisches', false);

INSERT INTO drucker (name, standort) VALUES ('EPSON', 'Budl');

INSERT INTO drucker\_artikel (dr\_id, art\_id) VALUES (2, 2);

INSERT INTO artikel\_bestellung (art\_id, bestellung\_id, bezahlt, zusatzlich) VALUES (2, 1, true, 'Strohhalm');

Statistik:

-Welcher Kellner wie viel Bestellungen gemacht hat

-Wie viel von welchem Artikel verkauft worden ist

UPDATE kellner SET anz\_bestellungen = anz\_bestellungen + 1 WHERE id = ?

UPDATE artikel SET anz\_bestellungen = anz\_bestellungen + 1 WHERE id = ?

Nächstes meeting:

Erklären was wir bis jetzt haben

Erklären was nicht möglich ist

Fragen:

Design?

Funktionalität?