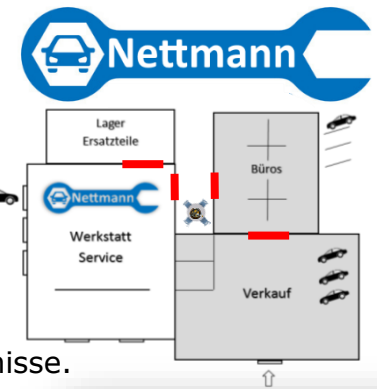


LS2.4 Zugangskontrolle – Tabellen erzeugen und Daten ändern (DDL/DML)

Neben den drei Tabellen Benutzer, Chips und Zutrittsversuche sollen nun auch die Kartenleser und spezielle Zugangsberechtigungen, welche an Zeitbereiche gebunden sind verwaltet werden. Hierzu müssen folgende Aufgaben anhand des SQL-Dumps **rfidvX_klein.sql** erledigt werden:



Beachten Sie:

- Loggen Sie Ihre Statements mit tee d:\xyz.log
- Ein erneuter Import von rfidvX_klein.sql löscht alle Ergebnisse.



Formulieren Sie für die folgenden Aufgaben die entsprechende SQL-Abfrage.

Nr.	Aufgabe
1	Legen Sie die Tabelle tblReader wie abgebildet an (ohne auto_increment!).
2	Legen Sie die Tabelle tblBerechtigung wie abgebildet an (ohne auto_increment!).
3	Ergänzen Sie, falls noch nicht beim Anlegen berücksichtigt, die Referenzielle Integrität der Beziehungen der neu angelegten Tabellen.
4	Überprüfen Sie im „Designer“ in PHP-MyAdmin ob alle Attribute, Schlüssel und Beziehungen wie abgebildet eingerichtet wurden. (Designer siehe Rückseite unten)
Wenn alles korrekt (<u>exakt</u> wie im logischen Model oben abgebildet, auch die Groß/Kleinschreibung!) eingerichtet wurde, importieren Sie bitte die Datei rfidvX_Datensaetze.sql, damit Sie bereits Beispieldatensätze in Ihren Tabellen haben.	
5	Fügen Sie einen neuen Mitarbeiter (bspw. Ihr Name) hinzu und überprüfen Sie die Einrichtung mit einer SQL Abfrage.
6	Fügen Sie für diesen neuen Mitarbeiter einen neuen Chip (mit sinnvollen Daten) hinzu und überprüfen Sie die Einrichtung mit einer SQL Abfrage.
7	Fügen Sie einen neuen Kartenleser hinzu und testen Sie im Simulator, ob (erstmal) niemand an diesem Zugang erhält.
8	Fügen Sie nun eine neue Berechtigung für diesen Chip hinzu und testen Sie den Zugang im Simulator.
9	Ändern Sie die Berechtigungen für Herrn Nettmann so ab, dass er nun an allen (auch an Ihrem neuen) Kartenlesern Zutritt erhält. (Testen Sie am Simulator)
10	Herr Tabor hat das Unternehmen verlassen, löschen Sie Ihn aus der Datenbank (Bedenken Sie, dass die Löschweitergabe deaktiviert ist. Sie müssen also sämtliche Abhängigkeiten von Herrn Tabor vorher löschen. Testen Sie am Simulator - Ab welchem Zeitpunkt war der Chip wirklich ohne Funktion?)
11	Frau Schmidt hat geheiratet. Ändern Sie Ihren Namen bitte in Schmidt-Huber und überprüfen Sie die Einrichtung mit einer SQL Abfrage.
12	Niemand außer Herr Nettmann soll Zugang zum Lager erhalten. Führen Sie alle nötigen Änderungen durch.

Inhalt von tblZutrittsversuche:

ZutrittsversuchID	Zeitstempel	tblChips_ChipsID	Ergebnis
2358	22.11.2017 12:27	12	Zutritt abgelehnt
2357	22.11.2017 12:26	12	Zutritt gestattet
2356	22.11.2017 12:26	12	Zutritt abgelehnt
2355	22.11.2017 12:25	12	Zutritt gestattet
2354	22.11.2017 12:25	12	Zutritt gestattet
2353	22.11.2017 12:25	12	Zutritt abgelehnt
2352	22.11.2017 12:25	12	Zutritt gestattet
2351	22.11.2017 12:25	12	Zutritt gestattet
2350	22.11.2017 12:25	12	Zutritt abgelehnt
2349	22.11.2017 12:25	12	Zutritt abgelehnt
...Datensätze gekürzt....			
2286	27.04.2017 09:56	5	Zutritt gestattet
2285	27.04.2017 09:56	5	Zutritt abgelehnt
2284	27.04.2017 09:56	5	Zutritt abgelehnt

Inhalt von tblChips:

ChipsID	ChipSerienNr	tblBenutzer_BenutzerID
2	01104A1CAAED	1
3	01104A6DBF89	2
4	01104A3EE085	3
5	010FA0DF2657	4
12	01104A3EC4A1	5
14	01104A41CBD1	6
15	011049BF7790	7

Inhalt von tblBenutzer:

BenutzerID	Nachname
1	Nettmann
2	Schmidt
3	Aydin
4	Tabor
5	Maier
6	Praktikant
7	Gast

Nach dem Import von rfidvX_Datensaeetze.sql:**Inhalt von tblBerechtigung:**

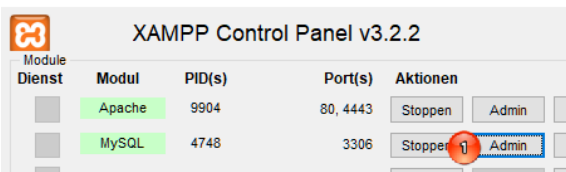
tblChips_ChipsID	tblReader_ReaderID
15	1
15	2
15	3
15	4
14	2
14	3
14	4
12	3
12	4
5	4
4	4

Inhalt von tblReader:

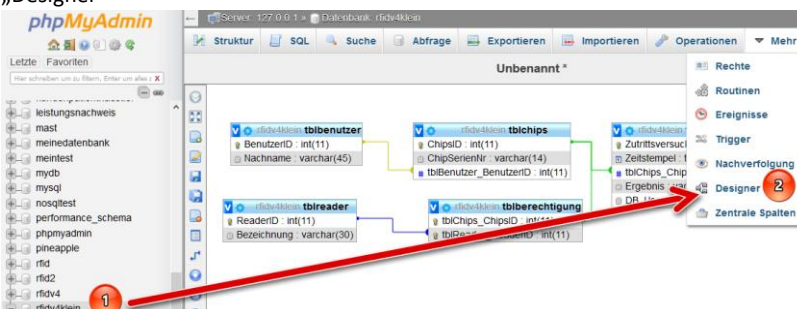
ReaderID	Bezeichnung
1	Eingang Lager
2	Eingang Werkstatt'
3	Eingang Büro
4	Eingang Büro/Verkauf

Tutorial: phpMyAdmin Designer-Öffnen.

1. Klicken Sie im XAMPP Control Panel auf die Schaltfläche MySQL Admin um phpMyAdmin zu öffnen



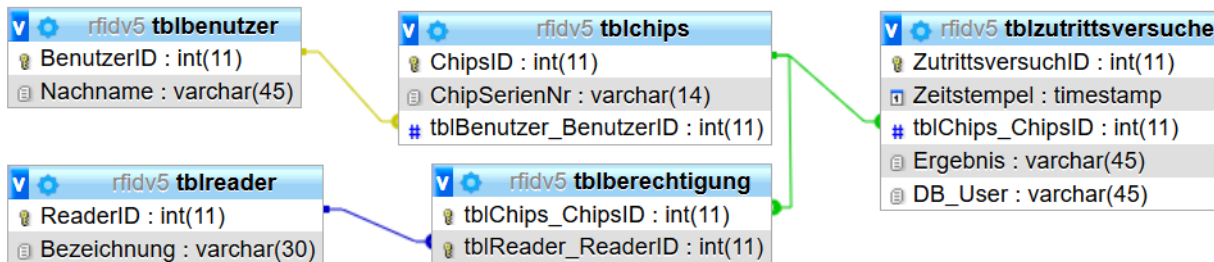
2. Wählen Sie auf der linken Seite die gewünschte Datenbank aus und klicken Sie anschließend auf den Reiter „Designer“



Zusatzaufgaben

Zugangskontrolle – Tabellen erzeugen und Daten ändern (DDL/DML)

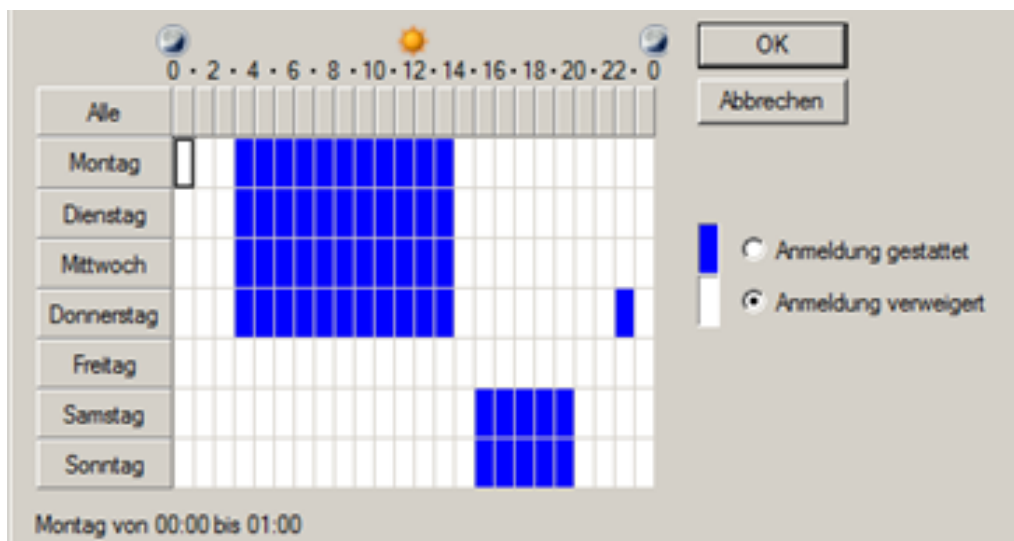
Neben den drei Tabellen tblBenutzer, tblChips und tblZutrittsversuche sollen nun auch die Kartenleser an den Türen und die Zugangsberechtigungen verwaltet werden.



1. Es soll nun festgehalten werden, mit welchem Chip wann an welcher Tür (reader!) ein Zutritt bzw. Zutrittsversuch erfolgte.
Erweitern Sie das obige Datenbankmodell und dann die Datenbank.

Ergänzung von ReaderID als FS in tblZutrittsversuche

2. Es soll nun eine Zeitabhängigkeit eingebaut werden.
Nicht immer sondern nur zu einer bestimmten Zeit soll ein Chip eine bestimmte Tür öffnen können. Für einen Chip können beliebig viele Zeitbereiche hinzugefügt werden.
Erweitern Sie zunächst das obige logische Datenbankmodell.



Bspw. durch neue Relation Zeitbereiche und der ZeitbereichsID als FS in der tblBerechtigung-Tabelle.

Nr	Aufgabe	Lösungsvorschlag
1	Legen Sie die Tabelle tblreader wie abgebildet an.	<pre>CREATE TABLE tblReader (ReaderID int(11) NOT NULL, Bezeichnung varchar(30), CONSTRAINT Prim_S_tblReader PRIMARY KEY (ReaderID));</pre>
2	Legen Sie die Tabelle tblBerechtigung wie abgebildet an.	<pre>CREATE TABLE tblBerechtigung (tblChips_ChipsID int(11) NOT NULL, tblReader_ReaderID int(11) NOT NULL, CONSTRAINT Prim_S_tblBerechtigung PRIMARY KEY (tblChips_ChipsID, tblReader_ReaderID));</pre>
3	Ergänzen Sie, falls noch nicht beim Anlegen berücksichtigt, die Referenzielle Integrität der Beziehungen der neu angelegten Tabellen.	<pre>ALTER TABLE tblBerechtigung ADD CONSTRAINT Fremd_S_tblReader FOREIGN KEY (tblReader_ReaderID) REFERENCES tblReader(ReaderID), ADD CONSTRAINT Fremd_S_tblChips FOREIGN KEY (tblChips_ChipsID) REFERENCES tblChips(ChipsID);</pre>
4	Überprüfen Sie im „Designer“ in PHP-MyAdmin ob alle Attribute, Schlüssel und Beziehungen wie abgebildet eingerichtet wurden.	
Wenn alles korrekt (<u>exakt</u> wie im logischen Model oben abgebildet, auch die Groß/Kleinschreibung!) eingerichtet wurde, importieren Sie bitte die Datei rfidvX_Datensaetze.sql, damit Sie bereits Beispieldatensätze in Ihren Tabellen haben.		
5	Fügen Sie einen neuen Mitarbeiter (bspw. Ihr Name) hinzu und überprüfen Sie die Einrichtung mit einer SQL Abfrage.	<pre>Insert into tblBenutzer (Nachname) values ('Romoningbakl'); --BenutzerID muss nicht vergeben werden, da auto_increment gesetzt ist</pre>
6	Fügen Sie für diesen neuen Mitarbeiter einen neuen Chip (mit sinnvollen Daten) hinzu und überprüfen Sie die Einrichtung mit einer SQL Abfrage.	<pre>--Ermittlung der ID von angelegten Benutzer → ID 9 Insert into tblChips (ChipSerienNr, tblBenutzer_BenutzerID) values ('Rad3151212', 9); --ChipsID muss nicht vergeben werden, da auto_increment ist --Ermittlung der letzten automatischen ID: Select last_insert_ID();</pre>
7	Fügen Sie einen neuen Kartenleser hinzu und testen Sie im Simulator, ob (erstmal) niemand an diesem Zugang erhält.	<pre>Insert into tblReader (ReaderID, Bezeichnung) values (5, 'Ausgang Lager');</pre>
8	Fügen Sie nun eine neue Berechtigung für diesen Chip hinzu und testen Sie den Zugang im Simulator.	<pre>Insert into tblBerechtigung (tblChips_ChipsID, tblReader_ReaderID) values (17, 5); -- Werte müssen ersetzt werden</pre>
9	Ändern Sie die Berechtigungen für Herrn Nettmann so ab, dass er nun an allen (auch an Ihrem neuen) Kartenlesern Zutritt erhält. (Testen Sie am Simulator)	<pre>Insert into tblBerechtigung (tblChips_ChipsID, tblReader_ReaderID) values (2, 1), (2, 2), (2, 3), (2, 4), (2, 5); ##Bessere Lösung Insert into tblBerechtigung (tblChips_ChipsID, tblReader_ReaderID) Select 2, ReaderID From tblReader; ##Bessere Lösung, da ich die ChipID von Herrn Nettman nicht kennen muss. Insert into tblBerechtigung (tblChips_ChipsID, tblReader_ReaderID) select (Select ChipsID from tblChips inner join tblbenutzer on tblbenutzer_BenutzerID = benutzerid where Nachname = "Nettmann"), ReaderID from tblReader</pre>
10	Herr Tabor hat das Unternehmen verlassen, löschen Sie ihn aus der Datenbank (Bedenken Sie, dass die Löschweitergabe deaktiviert ist. Sie müssen also sämtliche Abhängigkeiten von Herrn Tabor vorher löschen. Testen Sie am	<pre>-- 1. Löschen der Berechtigung, so dass Chip unbrauchbar wird Delete from tblBerechtigung where tblChips_ChipsID=5; -- 2. Löschen der tblZutrittsversuche wegen RefInt</pre>

IFA/IFS 11

	<p>Simulator - Ab welchem Zeitpunkt war der Chip wirklich ohne Funktion?)</p>	<pre>Delete from tblZutrittsversuche where tblChips_ChipsID=5; -- 3. Löschen der Chips wegen RefInt Delete from tblChips where tblBenutzer_BenutzerID=4; -- 4. Löschen des Benutzers via ID Delete from tblBenutzer where BenutzerID=4;</pre>
11	<p>Frau Schmidt hat geheiratet. Ändern Sie Ihren Namen bitte in Schmidt-Huber und überprüfen Sie die Einrichtung mit einer SQL Abfrage.</p>	<pre>Update tblBenutzer set Nachname='Schmidt-Huber' where BenutzerID = 2;</pre>
12	<p>Niemand außer Herr Nettmann soll in Zukunft Zugang zum Lager erhalten. Führen Sie alle nötigen Änderungen durch.</p>	<pre>-- Laut DB hat Nettmann ChipID 2 und Lager ReaderID 1 Delete from tblBerechtigung where tblReader_ReaderID = 1 and tblChips_ChipsID != 2; --Alternative für != 2 wäre: Delete from tblBerechtigung where tblReader_ReaderID = 1 and tblChips_ChipsID != NOT IN (Select C.ChipsID from tblChips as C where C.tblBenutzer_BenutzerID = 1);</pre>