Freizeitpark mit Pension

Projektbeschreibung

Ausgeführt von: Hochwald Marius, Miljevic Stefan, Poljanc Alina, Wermann Josef  
Personenkennzeichen: 1810257159, 1810257073, 1810257149, 1810257182

Begutachter: Ing. Martin Docsek MSc, Mag. Friedrich Wottawa-Posch BSc.

Wien, 13.01.2019

# Projektinhalt

Das Projekt umfasst einen ein Verwaltungssystem für einen Freizeitpark. Dieser Park beinhaltet Personal wie zum Beispiel Geschäftsführung, Büro, Schauspieler, Security, Ticketing oder auch Putzpersonal und auch Attraktionen, also Riesenrad, Kinderkarussell, ein Gruselhaus und weitere Attraktionen. Ebenfalls ist eine Pension Teil des Freizeitparkes, diese hat eine bestimmte Anzahl an Zimmer in unterschiedlichen Kategorien mit unterschiedlichen Preisen und Bettenzahlen. Die Gäste können darüber hinaus über eine Pensionsform entscheiden (Vollpension, Halbpension, Tageskarte, je für Kinder und Erwachsene).

Das Verwaltungssystem ermöglicht:

* Check In

Ein neuer oder schon im System vorhandener Gast kann eingecheckt werden. Dabei kann er die Raumkategorie und Pensionsform wählen. Mit den angegebenen Informationen zusätzlich zu dem Ankunftsdatum und der Anzahl der Nächte wird ein passendes Zimmer gesucht, dem Gast dieses Zimmer zugewiesen und eine Rechnung erstellt.

Des Weiteren kann man einer Buchung bzw. einer Rechnung weitere Gäste hinzufügen solange noch genügend Betten in den gebuchten Zimmern verfügbar sind.

Für den Fall, dass ein Gast noch ein weiteres Zimmer zu einer Buchung hinzufügen will ist dies auch ermöglicht.

* Rechnungsdarlegung

Die Buchungen mit dem jeweiligen Preis können im System dargelegt werden. Die Rechnung umfasst somit den Gesamtpreis für die gewählte Pensionsform und den Preis für die hinzugefügten Zimmer in der angegebenen Kategorie für den gewählten Zeitraum und alle an der Rechnung vermerkten Gäste.

* Gästeübersicht

Eine Übersicht der eingecheckten Gäste liefert alle notwendigen Informationen: Vorname und Nachname des Gastes, Anreise- und Abreisedatum, gewählte Pensionsform und die Zimmernummer(n) auf dem der Gast verbucht ist.

* Zimmerverfügbarkeit

Anzeige vorhandener Zimmer der Pension, Zimmer gelistet nach Zimmernummer und je Zimmer Beschreibung (Zimmerkategorie), Preis und Status (frei oder nächstes Ausreisedatum).

* Personalverwaltung

Zu dem eingefügten Personal wird Vor- und Nachname sowie Gender angezeigt, Abteilung (Geschäftsführung, Büro, etc.), Gehalt und Attraktion können hier verwaltet werden.

# Projektumgebung

Das Projekt wurde in zwei wesentliche Teile gegliedert, die Datenbank und den Webserver. Die Datenbank läuft in der vorkonfigurierten Open Suse Oracle VM von Moodle, der vorhandene User *system* wird dafür genützt. Zum erstellen des Tablespaces, der Tabellen, Indizes, Trigger, Prozeduren usw. in PL/SQL wurde eine Verbindung zur VM über die SQL Workbench aufgebaut.

Der Webserver läuft lokal z.B. auf XAMPP wobei die GUI-Programmiersprache PHP ist. Zusätzlich wird eine Oracle Schnittstelle (oci-8) genützt, dazu musste der Oracle instant Client installiert werden. Diese Schnittstelle unterstützt SQL und PL/SQL Statements sowie weitere Funktionen wie das binden von PHP-Variablen zu Oracle placeholders.

# Umsetzung

## Datenbank

### Datenbankschema

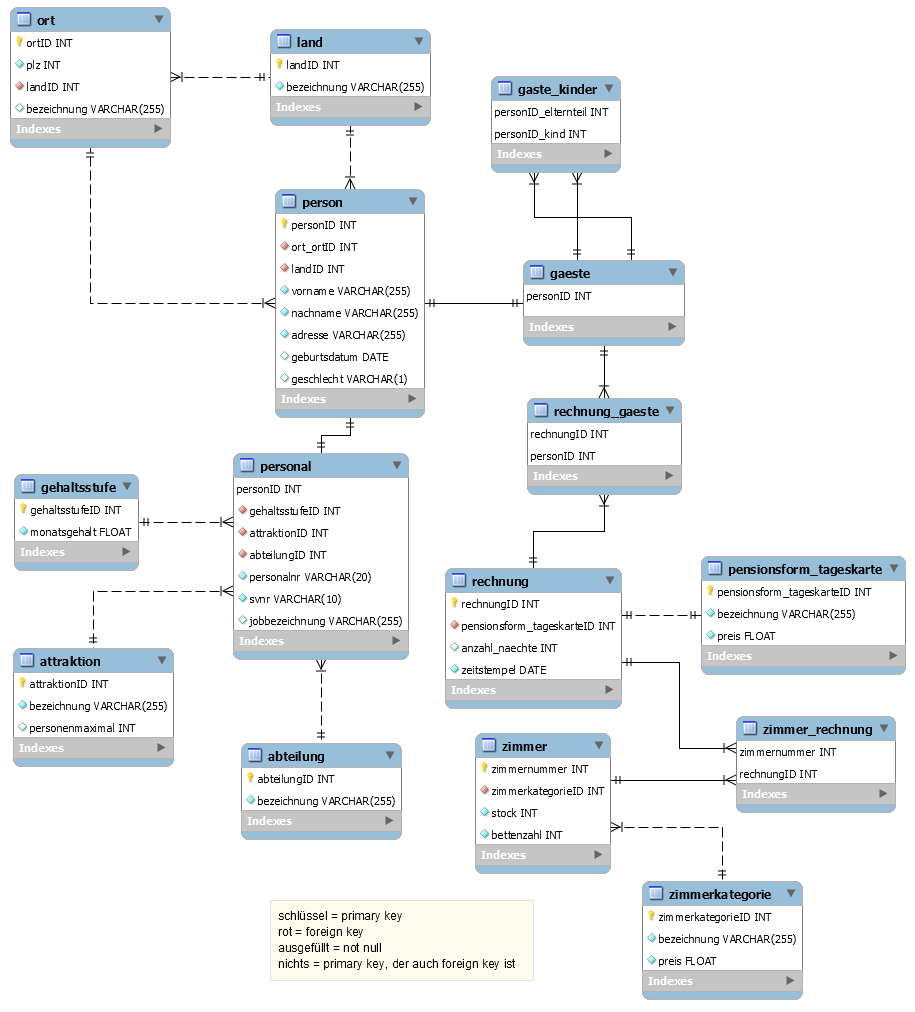


Abbildung 1: Datenbankschema

### Tabellen

Tabelle 1: Tabellenübersicht

|  |  |  |
| --- | --- | --- |
| Name | Inhalt | Indizes |
| land | Laendernamen und LaenderIDs | / |
| ort | Ortnamen mit zugehöriger PLZ und Land | ind\_ort |
| person | generelle Personenübersicht mit Vorname, Nachname, Geburtsdatum, Geschlecht Adresse, Ort und Land | ind\_person, ind\_person\_land |
| gehaltsstufe | Zuordnung Gehaltsstufen und Monatsgehalt | / |
| attraktion | Attraktionen mit maximaler gleichzeitiger Personenanzahl | / |
| abteilung | Abteilungsbezeichnungen | / |
| personal | Personalzuordnung zu Gehalt, eventuell Attraktion Personalnummer, SVNR und Abteilung | ind\_personal |
| gaeste | Definition von Personen als Gästen | / |
| gaeste\_kinder | Hilfstabelle Kinder zu Eltern zuordnen | / |
| pensionsform\_tageskarte | Bezeichnungen von möglichen Karten und Pensionsformen inkl. Preis | / |
| zimmerkategorie | Bezeichnungen von Zimmerkategorien inkl. Preis | / |
| zimmer | Zimmernummern mit Zuordnung zu Kategorie, Stock und Bettenzahl | ind\_zimmer |
| rechnung | Gesamtrechnungen mit Karte/Pension, Nächtezahl und Zeitstempel | ind\_rechnung |
| zimmer\_rechnung | Hilfsstabelle für Rechnung des Zimmers | / |
| rechnung\_gaeste | Hilfsstabelle um Rechnung dem Gast zuzuweisen | / |

### Spalten

Tabelle 2: Spaltenbeschreibung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tabelle | Spaltenname | Datentyp | Constraints | Inhalt |
| land | landID | INT | PRIMARY KEY | Eindeutige Identifizierung eines Landes |
| bezeichnung | VARCHAR(255) | UNIQUE NOT NULL | Bezeichnung des Landes |
| ort | ortID | INT | PRIMARY KEY | Eindeutige Identifizierung eines Ortes |
| plz | INT | NOT NULL | Postleitzahl |
| landID | INT | REFERENCES land(landID) ON DELETE SET NULL | ID des Landes dem der Ort angehört |
| bezeichnung | VARCHAR(255) |  | Bezeichnung des Ortes |
| person | personID | INT | PRIMARY KEY | Eindeutige Identifizierung einer Person |
| landID | INT | REFERENCES land(landID) ON DELETE SET NULL | ID des Landes der die Person angehört |
| vorname | VARCHAR(255) | NOT NULL | Vorname der Person |
| nachname | VARCHAR(255) | NOT NULL | Nachname der Person |
| adresse | VARCHAR(255) | NOT NULL | Adresse der Person |
| ortID | INT | REFERENCES ort(ortID) ON DELETE SET NULL | ID des Ortes der die Person angehört |
| geburtsdatum | DATE | / | Geburtsdatum der Person |
| geschlecht | VARCHAR(1) | / | Geschlecht der Person (m/w) |
| gehaltsstufe | gehaltsstufeID | INT | PRIMARY KEY | Eindeutige Identifizierung einer Gehaltsstufe |
| monatsgehalt | FLOAT | / | Monatsgehalt in Euro |
| attraktion | attraktionID | INT | PRIMARY KEY | Eindeutige Identifizierung einer Attraktion |
| bezeichnung | VARCHAR(255) | NOT NULL | Name der Attraktion |
| personenmaximal | INT | / | Maximale Anzahl an Personen die diese Attraktion gleichzeitig nützen können |
| abteilung | abteilungID | INT | PRIMARY KEY | Eindeutige Identifizierung einer Abteilung |
| bezeichnung | VARCHAR(255) | / | Bezeichnung der Abteilung |
| personal | personID | INT | PRIMARY KEY REFERENCES person(personID) ON DELETE CASCADE | ID der Person die als Personal vermerkt ist |
| gehaltsstufeID | INT | REFERENCES gehaltsstufe(gehaltsstufeID) ON DELETE SET NULL | ID der Gehaltsstufe in der sich das Personal befindet |
| attraktionID | INT | REFERENCES attraktion(attraktionID) ON DELETE SET NULL | ID der Attraktion der dem Personal zugeordnet ist |
| personalnr | VARCHAR(20) | UNIQUE NOT NULL | Personalnummer |
| svnr | VARCHAR(10) | UNIQUE NOT NULL | Sozialversicher-  ungsnummer |
| abteilungID | INT | REFERENCES abteilung(abteilungID) ON DELETE SET NULL | ID der Abteilung der dem Personal zugeordnet ist |
| gaeste | personID | INT | PRIMARY KEY REFERENCES person(personID) ON DELETE CASCADE | ID der Person die als Gast hinzugefügt wurde |
| gaeste\_kinder | personID\_elternteil | INT | REFERENCES gaeste(personID) ON DELETE CASCADE |  |
| personID\_kind | INT |  |  |
|  |  | REFERENCES gaeste(personID) ON DELETE CASCADE |  |
| pensionsform\_tageskarte | pensionsform\_tageskarteID | INT | PRIMARY KEY | Eindeutige Identifizierung einer Pensionsform |
| bezeichnung | VARCHAR(255) | NOT NULL | Bezeichnung der Pensionsform |
| preis | FLOAT | NOT NULL | Preis der Pensionsform |
| zimmerkategorie | zimmerkategorieID | INT | PRIMARY KEY | Eindeutige Identifizierung der Zimmerkategorie |
| bezeichnung | VARCHAR(255) | NOT NULL | Bezeichnung der Zimmerkategorie |
| preis | FLOAT | NOT NULL | Preis der Zimmerkategorie |
| zimmer | zimmernummer | INT | PRIMARY KEY | Eindeutige Zimmernummer |
| zimmerkategorieID | INT | REFERENCES zimmerkategorie(zimmerkategorieID) ON DELETE SET NULL | ID der Zimmerkategorie die dem Zimmer zugeordnet ist |
| stock | INT | NOT NULL | Stock in dem sich das Zimmer befindet |
| bettenzahl | INT | NOT NULL | Anzahl der Betten des Zimmers |
| rechnung | rechnungID | INT | PRIMARY KEY | Eindeutige Identifizierung einer Rechnung |
| pensionsform\_tageskarteID | INT | REFERENCES pensionsform\_tageskarte(pensionsform\_tageskarteID) ON DELETE SET NULL | ID der angegebenen Pensionsform des Aufenthaltes |
| anzahl\_naechte | INT | / | Anzahl der Nächte des Aufenthaltes |
| zeitstempel | DATE | NOT NULL | Ankunftsdatum der Gäste |
| zimmer\_rechnung | zimmernummer | INT | REFERENCES zimmer(zimmernummer) ON DELETE CASCADE | Nummer des Zimmers |
| rechnungID | INT | REFERENCES rechnung(rechnungID) ON DELETE CASCADE | ID der Rechnung |
| rechnung\_gaeste | rechnungID | INT | REFERENCES rechnung(rechnungID) ON DELETE CASCADE | ID der Rechnung |
| personID | INT | REFERENCES gaeste(personID) ON DELETE CASCADE | ID des Gastes |

### Trigger

#### age\_Trigger

Art: Before insert

Zweck: Überprüft beim Eintragen der Person, ob das Datum realistisch nachvollziehbar ist.

Tabellen und Attribute: person.geburtsdatum

#### Zimmerk\_preischeck\_trigger

Art: Before insert

Zweck: Überprüft beim Eintragen eines Preises, dass dieser nicht irrtümlich negativ eingetragen wird.

Tabellen und Attribute: zimmerkategorie.preis

#### CheckInLog\_trigger

Art: After insert

Zweck: Loggt alle Checkins, die durch die CheckIn Prozedur duchgeführt werden.

Tabellen und Attribute: rechnunggaeste.rechnungID; rechnunggaeste.personID; person.vorname; person.nachname

#### Zimmerlog\_trigger

Art: After Insert

Zweck: Erhöht einen Counter in der LogginTable Zimmerlog jedes Mal um 1, wenn ein Zimmer gebucht wurde.

Tabellen und Attribute: zimmer\_rechnung.zimmernummer

**Packages / Stored Procedures / Funktionen**

#### freizeitpark

Art: TABLESPACE

Zweck: Eigenen Tablespace für das Projekt, um alle weiteren Objekte darin zu kapseln.

#### t\_varchar2\_tab[[1]](#footnote-1)

Art: TYPE

Zweck: Tabelle aus Zeichenketten (VARCHAR2(400)) um beim gruppieren der View gaeste\_view\_group die Zimmernummern als einen String mit Komma getrennt darzustellen.

#### tab\_to\_string1

Art: FUNCTION

Zweck: Hilfsfunktion um beim Gruppieren der View gaeste\_view\_group die Zimmernummern als einen String mit Komma getrennt darzustellen. Fasst eine Tabelle von Strings als einen String zusammen.

Eingangsvariablen:

* t\_varchar2\_tab t\_varchar2\_tab
* p\_delimite VARCHAR2 DEFAULT ‘,’

Ausgangsvariablen:

* VARCHAR2

#### seq\_rechnung

Art: SEQUENCE

Zweck: Sequenz für den Primärschlüssel (rechnungID) der Tabelle rechnung.

#### sp\_checkin

Art: PROCEDURE

Zweck: Die Prozedur erstellt einen Gast, wenn er noch nicht existiert, eine Rechnung bzw. Buchung und reserviert ein Zimmer für den angegebenen Zeitraum, Pensionsform und Zimmerkategorie. Im Falle, dass der übergebene Name nicht in der Tabelle person enthalten ist , kein Zimmer in der gewünschten Kategorie frei ist oder die Person ist bereits auf einem anderem Zimmer zur gleichen Zeit gebucht, wird eine Fehlermeldung ausgegeben.

Eingangsvariablen:

* l\_v\_nachname\_in VARCHAR - Nachname des Gastes.
* l\_v\_vornamen\_in VARCHAR - Vorname des Gastes.
* l\_d\_zeitstempel\_in VARCHAR - Ankunftdatum des Gastes.'YYYY-MM-DD'
* l\_i\_anzahl\_naechte\_in INTEGER - Anzahl der Nächte des Aufenhaltes.
* l\_v\_bezeichnung\_in VARCHAR - Bezeichnung der gewünschten Zimmerkategorie.
* l\_v\_bezeichnung\_form\_in VARCHAR - Bezeichnung der gewüschten Pensionsform

Ausgangsvariablen:

Keine 🡪 im Falle von Fehlern werden verschiedene Fehlermeldungen (mithilfe von raise\_application\_error(errno,errmsg)) geworfen.

#### sp\_checkin\_add

Art: PROCEDURE

Zweck: Die Prozedur erstellt einen Gast, wenn er noch nicht existiert und fügt ihm der Rechnung des angegebenen primären Gastes für das angegebene Ankunftsdatum hinzu. Im Falle, dass der übergebene Name nicht in der Tabelle person enthalten ist, kein Bett in den Zimmern der primären Buchung über ist, oder die Person bereits zur gleichen Zeit ein anderes Zimmer gebucht hat, wird eine Fehlermeldung ausgegeben.

Eingangsvariablen:

* l\_v\_nachname\_in VARCHAR - Nachname des Gastes.
* l\_v\_vornamen\_in VARCHAR - Vorname des Gastes.
* l\_d\_zeitstempel\_in VARCHAR - Ankunftdatum des Gastes.'YYYY-MM-DD'
* l\_v\_nachname\_primary\_in VARCHAR - Nachname zu dem der Gast hinzugebucht wird
* l\_v\_vornamen\_primary\_in VARCHAR - Vorname zu dem der Gast hinzugebucht wird.

Ausgangsvariablen:

Keine 🡪 im Falle von Fehlern werden verschiedene Fehlermeldungen (mithilfe von raise\_application\_error(errno,errmsg)) geworfen.

#### sp\_checkin\_add\_room

Art: PROCEDURE

Zweck: Die Prozedur fügt ein neues Zimmer in der gewünschten Kategorie der Rechnung des angegebenen primären Gastes für das angegebene Ankunftsdatum hinzu. Im Falle, dass der übergebene Name nicht in der Tabelle person enthalten ist oder kein Zimmer in der gewünschten Kategorie in dem angegebenen Zeitraum über ist, wird eine Fehlermeldung ausgegeben.

Eingangsvariablen:

* l\_d\_zeitstempel\_in VARCHAR - Ankunftdatum des Gastes.'YYYY-MM-DD'
* l\_v\_nachname\_primary\_in VARCHAR - Nachname zu dem das Zimmer hinzugebucht wird
* l\_v\_vornamen\_primary\_in VARCHAR - Vorname zu dem das Zimmer hinzugebucht wird.
* l\_v\_bezeichnung\_in VARCHAR - Bezeichnung der gewünschten Zimmerkategorie.

Ausgangsvariablen:

Keine 🡪 im Falle von Fehlern werden verschiedene Fehlermeldungen (mithilfe von raise\_application\_error(errno,errmsg)) geworfen.

**sp\_calculate\_invoice\_price**

Art: PROCEDURE

Zweck: Die Prozedur berechnet den endgültigen Betrag einer Rechnung, wobei die ID einer Rechnung übergeben wird. Im Falle eines Fehlers wird NULL in der OUT Variable zurückgegeben.

Eingangsvariablen:

* l\_i\_invoiceID\_in INT - ID der Rechnung

Ausgangsvariablen:

* l\_n\_price\_ou NUMBER - Preis der Rechnung bzw. Buchung oder, im Fehlerfall, NULL

**sp\_invoice\_guest\_list**

Art: PROCEDURE

Zweck: Die Prozedur retourniert alle Namen der Gäste einer Buchung zu einer übergebenen rechnungID. Wenn keine Daten vorhanden sind, wird NULL in der OUT Variable zurückgegeben.

Eingangsvariablen:

* l\_i\_invoiceID\_in INT - ID der Rechnung

Ausgangsvariablen:

* l\_v\_guestList\_ou VARCHAR2 – Gäste zugehörig zu der Rechnung bzw. Buchung oder, im Fehlerfall, NULL

#### PERSONAL\_OBJ\_TYPE

Art: TYPE

Zweck: Objekt aus personID, vorname, nachname, geschlecht, abteilungID, gehaltsstufeID unf attraktionID.

#### PERSONAL\_TAB\_TYPE

Art: TYPE

Zweck: Tabelle bestehend aus Objekten des Types PERSONAL\_OBJ\_TYPE.

#### PERSONAL\_OBJ\_TYPE + PERSONAL\_TAB\_TYPE

🡪 generelle Personalübersicht mit PersonID, Vorname, Nachname, Geschlecht, AbteilungID, GehaltsstufeID und AttraktionID

**f\_getPersonalData\_personal\_type**

Art: FUNCTION

Zweck: Die Funktion retourniert Personal-relevanten Daten zur Anzeige im Webclient.

Eingangsvariablen:

keine

Ausgangsvariablen:

* l\_personal\_tab\_type\_details PERSONAL\_TAB\_TYPE – Tabelle mit allen relevanten Informationen des Personals, Spalten siehe PERSONAL\_OBJ\_TYPE.

**sp\_updatePersonalData**

Art: PROCEDURE

Zweck: Die Prozedur aktualisiert die Personaldaten.

Eingangsvariablen:

* l\_i\_personID\_in INT - personID des Personals
* l\_i\_newAbteilungID\_in INT - ID der neuen Abteilung
* l\_i\_newGehaltsstufeID\_in INT - ID der neuen Gehaltsstufe
* l\_i\_newAttraktionID\_in INT - ID der neu zugewiesenen Attraktion

Ausgangsvariablen:

Keine 🡪 im Falle von Fehlern werden verschiedene Fehlermeldungen ausgegeben.

**sp\_room\_status**

Art: PROCEDURE

Zweck: Eine Prozedur die überprüft ob das Zimmer, übergeben mithilfe der Zimmernummer, derzeit leer steht. Falls das Zimmer leer ist, dann wird über die Variable l\_v\_result\_ou NULL zurückgegeben. Andernfalls wird das Abreisedatum der letzten Buchung zurückgegeben.

Eingangsvariablen:

* l\_i\_roomNr\_in INT - Zimmernummer des Zimmers dessen Status abgefragt wird

Ausgangsvariablen:

* l\_v\_result\_ou VARCHAR2 – Letztes Abreisedatum oder NULL = Status des Zimmers

**f\_is\_room\_free**

Art: FUNCTION

Zweck: Eine Funktion die einen booleschen Wert retourniert, basierend auf den Status eines Zimmers (Besetzt oder frei)

Eingangsvariablen:

* l\_i\_room\_number\_in INTEGER - Zimmernummer des Zimmers, dessen Status abgefragt wird
* l\_d\_date\_in DATE - Startdatum des abzufragenden
* l\_i\_nights\_number\_in INTEGER - Anzahl der Nächte, d.h. Startdatum + Nächte geben den Zeitraum für die Statusabfrage an

Ausgangsvariablen:

* BOOLEAN – TRUE, wenn Zimmer frei; FALSE, wenn Zimmer belegt

**Views**

**personal\_view**

Inhalt: Der View zeigt alle Personalmitglieder mit Personalnummer, Nachname, Vorname, Alter, Geschlecht, Gehalt, Abteilungsbezeichnung und falls vorhanden zugewiesener Attraktion an.

Basisrelationen:

* person ↔ personal
* personal ↔ gehaltsstufe
* personal ↔ abteilung
* personal ↔ attraktion

**kinder\_view**

Inhalt: Der View zeigt bei Gaesten zu allen Kindern die dazugehörigen Eltern an (bzw., falls nur ein Elternteil bekannt ist, nur dieses.)

Basisrelationen:

* gaeste\_kinder ↔ gaeste
* gaeste ↔ person

**zimmer\_view**

Inhalt: Der View zeigt die Zimmerinformationen (nummer, bezeichnung und preise)

Basisrelationen:

* zimmer ↔ zimmerkategorie

**rechnung\_view**

Inhalt: Der View zeigt die Basis Rechnungsinformationen (rechnungsID, zeitstempel, anzahl der naechte und pensionsform bezeichnung). Dieser View wird für Darstellung der Rechnungsdaten verwendet.

Basisrelationen:

* rechnung ↔ pensionsform\_tageskarte

**gaeste\_view**

Inhalt: Der View zeigt alle für die Gäste relevanten Daten: Vor- und Nachname eines Gastes, Zimmernummer auf dem der Gast eingecheckt ist, die Zimmerkategorie und gebuchte Pensionsform, sowie Ankunfts- und Abreisedatum.

Basisrelationen:

* person ↔ gaeste
* gaeste ↔ rechnung\_gaeste
* rechnung\_gaeste ↔ rechnung
* rechnung ↔ zimmer\_rechnung
* zimmer\_rechnung ↔ zimmer
* zimmer ↔ zimmerkategorie
* rechnung ↔ pensionsform\_tageskarte

**gaeste\_view\_grouped**

Inhalt: Der View zeigt alle für die Gäste relevanten Daten gruppiert nach Buchung(von-bis) sowie Vor- und Nachname: Vor- und Nachname eines Gastes, Zimmernummer auf dem der Gast eingecheckt ist und gebuchte Pensionsform, sowie Ankunfts- und Abreisedatum.

Basisrelationen:

* person ↔ gaeste
* gaeste ↔ rechnung\_gaeste
* rechnung\_gaeste ↔ rechnung
* rechnung ↔ zimmer\_rechnung
* zimmer\_rechnung ↔ zimmer
* zimmer ↔ zimmerkategorie
* rechnung ↔ pensionsform\_tageskarte

**Indizes**

**ind\_person**

Tabelle: person

Attribut(e): nachname, vorname

**ind\_person\_land**

Tabelle: person

Attribut(e): landID, ortID

**ind\_ort**

Tabelle: ort

Attribut(e): landID

**ind\_personal**

Tabelle: personal

Attribut(e): abteilungID, attraktionID, gehaltsstufeID

**ind\_rechnung**

Tabelle: rechnung

Attribut(e): pensionsform\_tageskarte

**ind\_zimmer**

Tabelle: zimmer

Attribut(e): zimmerkategorieID, zimmernummer

## Webserver

### Ordnerstruktur

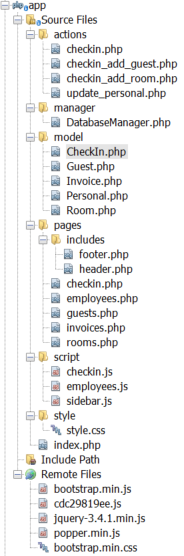


Abbildung 2: Ordnerstruktur Webserver

### Check In

Pfad: <http://localhost/pages/checkin.php>

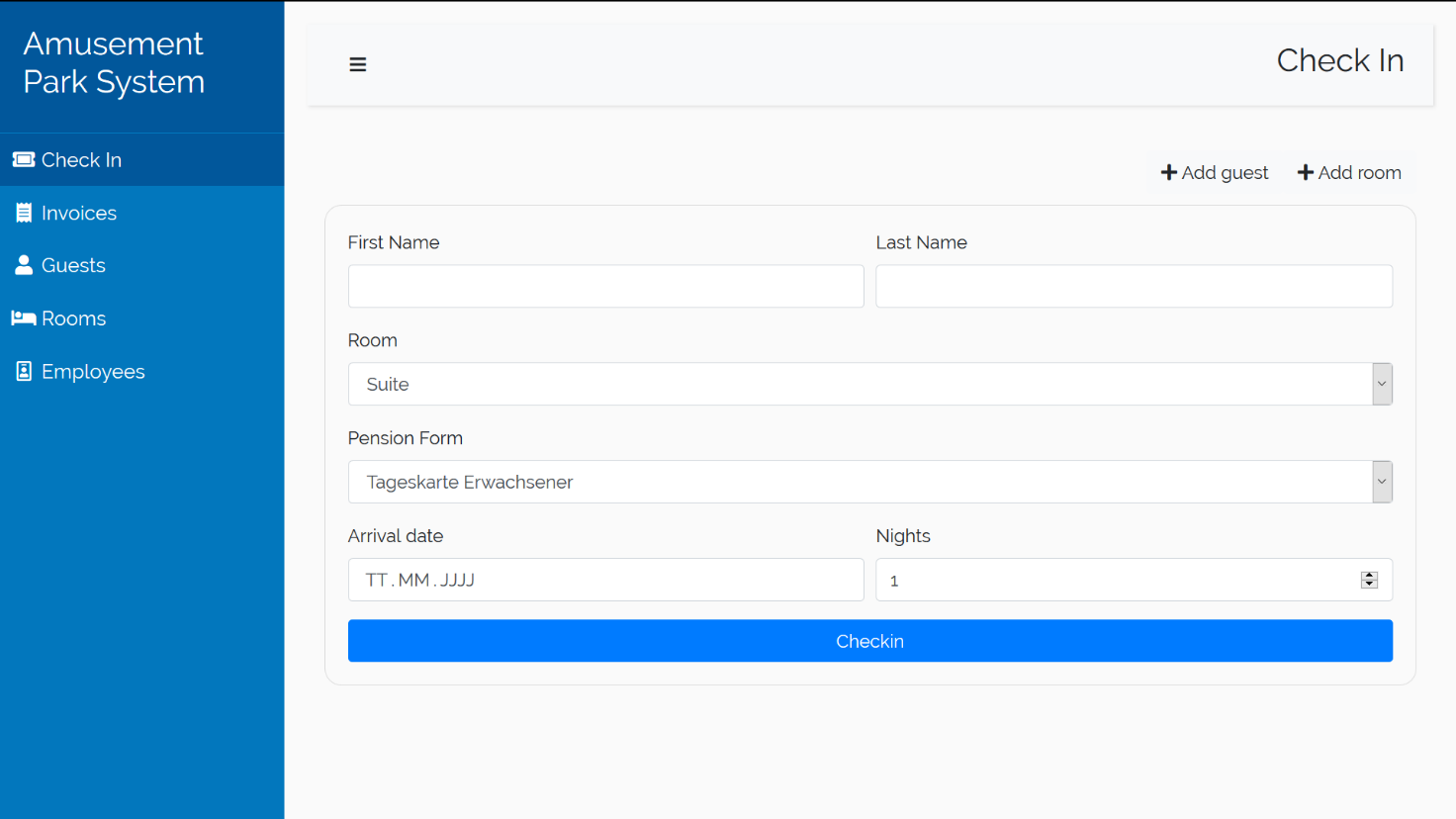


Abbildung 3: Check In

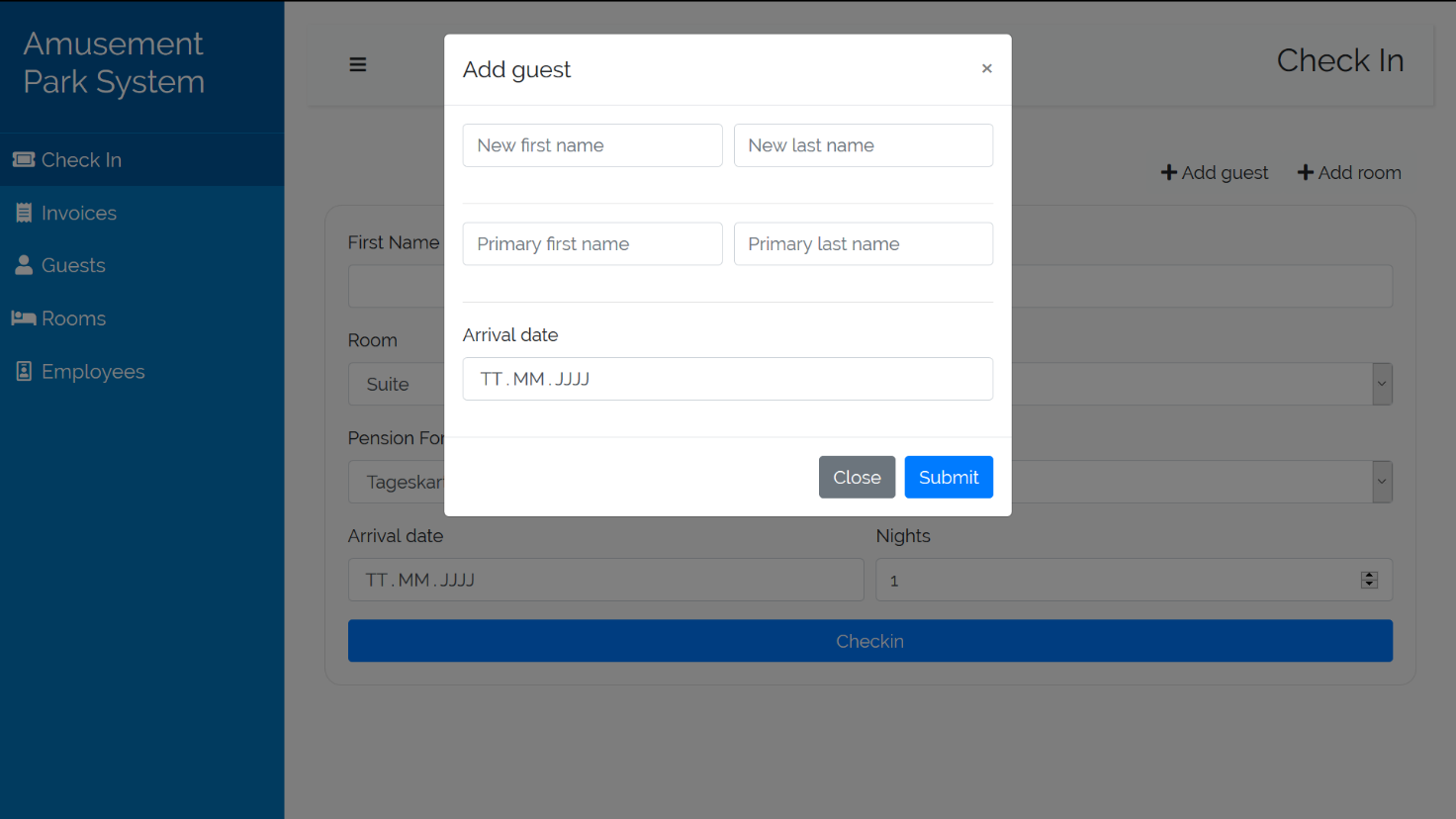


Abbildung 4: Gast zu Buchung hinzufügen

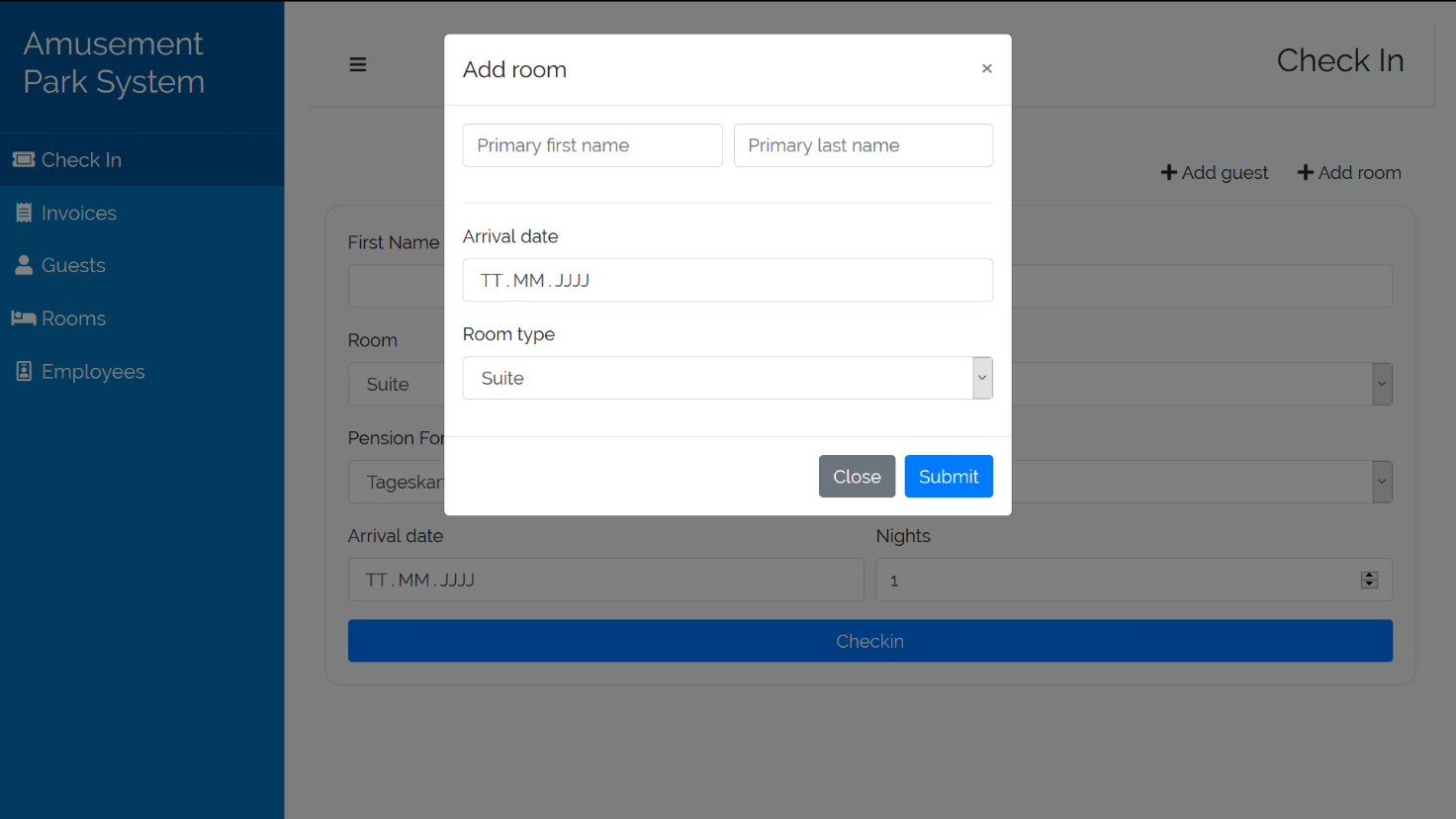


Abbildung 5: Zimmer zu Buchung hinzufügen

### Rechnungsdarlegung

Pfad: <http://localhost/pages/invoices.php>

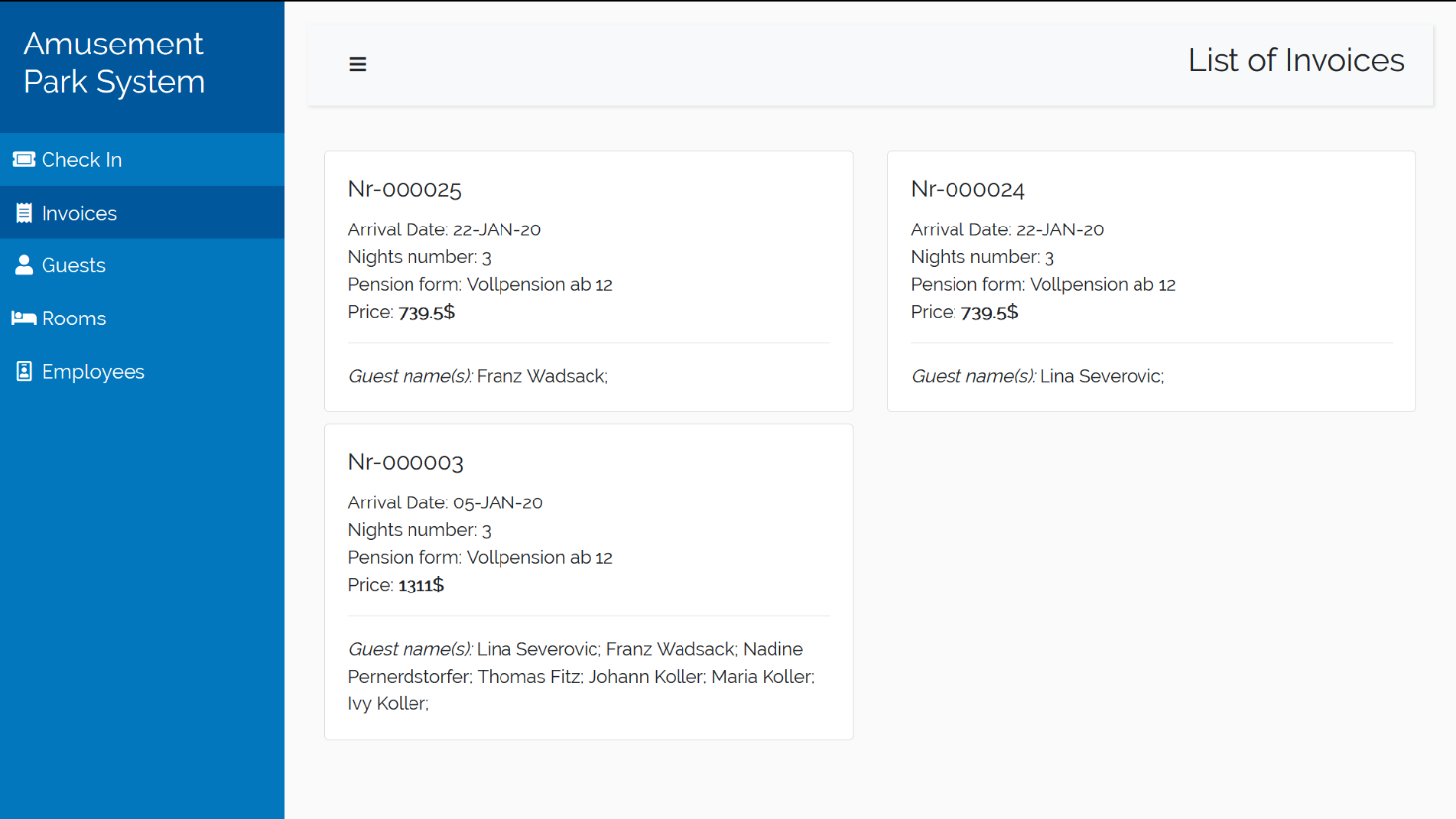


Abbildung 6: Rechnungsdarlegung

### Gästeübersicht

Pfad: <http://localhost/pages/guests.php>

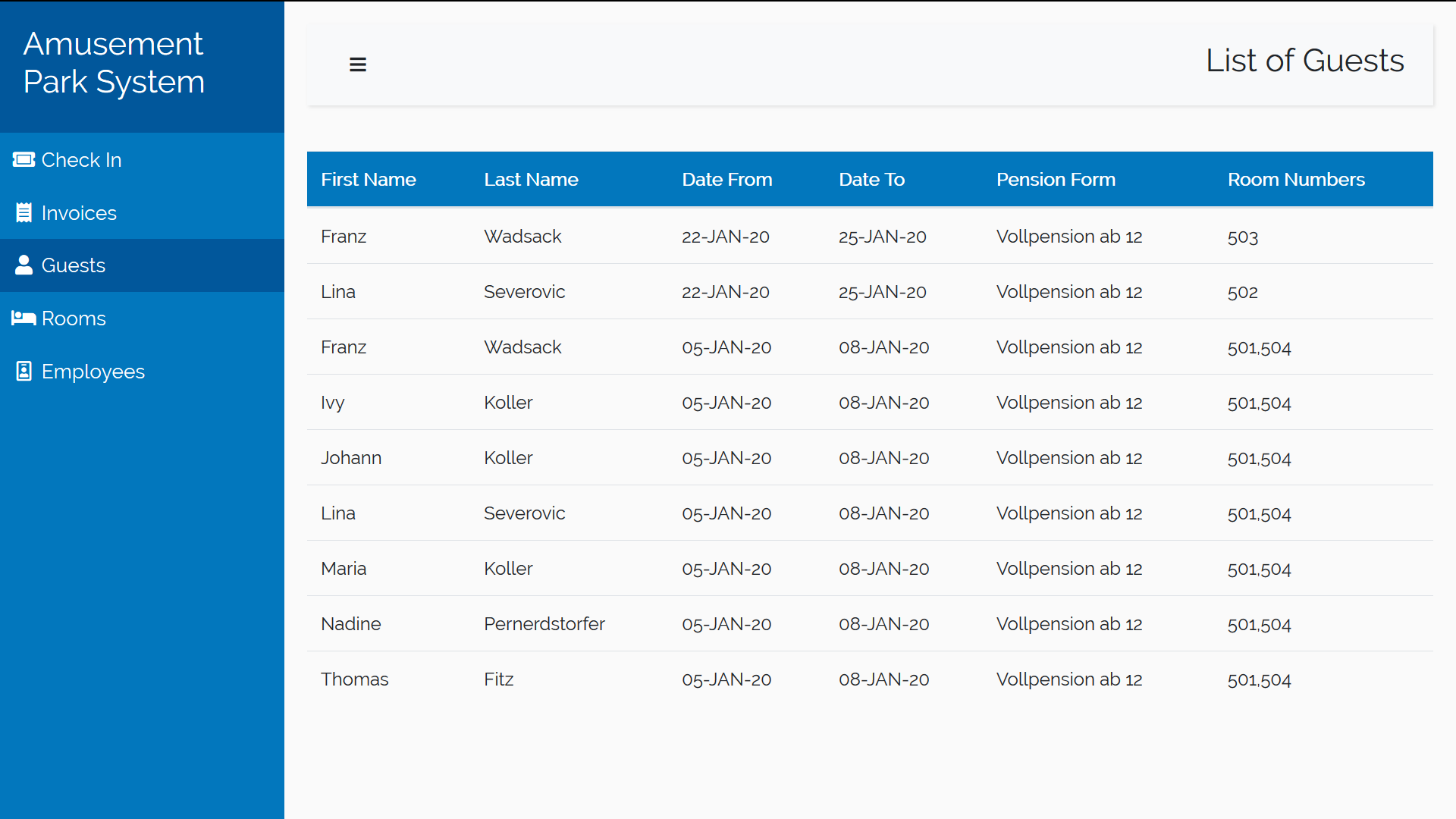


Abbildung 7: Gästeübersicht

### Zimmerverfügbarkeit

Pfad: <http://localhost/pages/rooms.php>

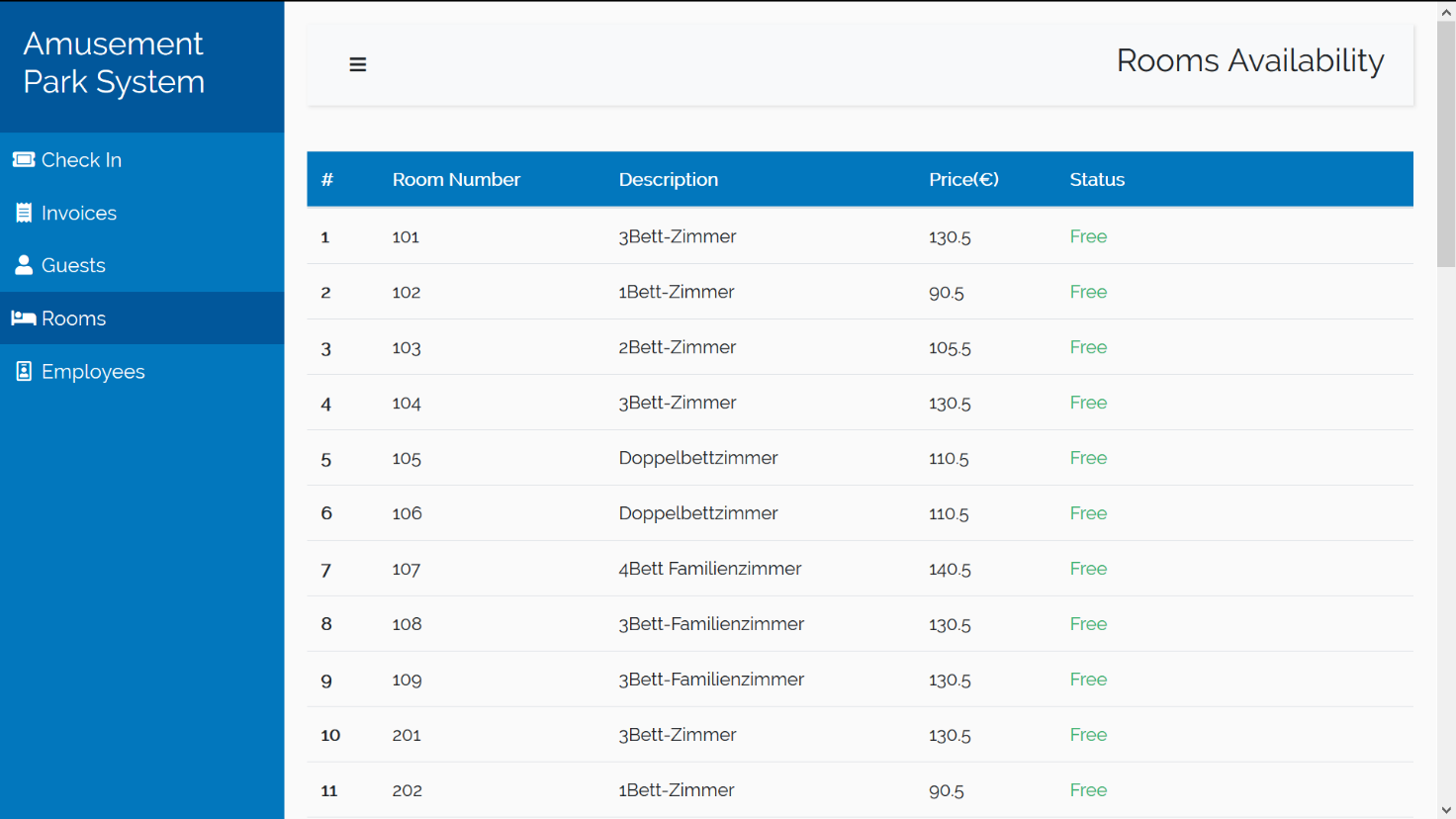


Abbildung 8: Zimmerverfügbarkeit

### Personalverwaltung

Pfad: <http://localhost/pages/employees.php>

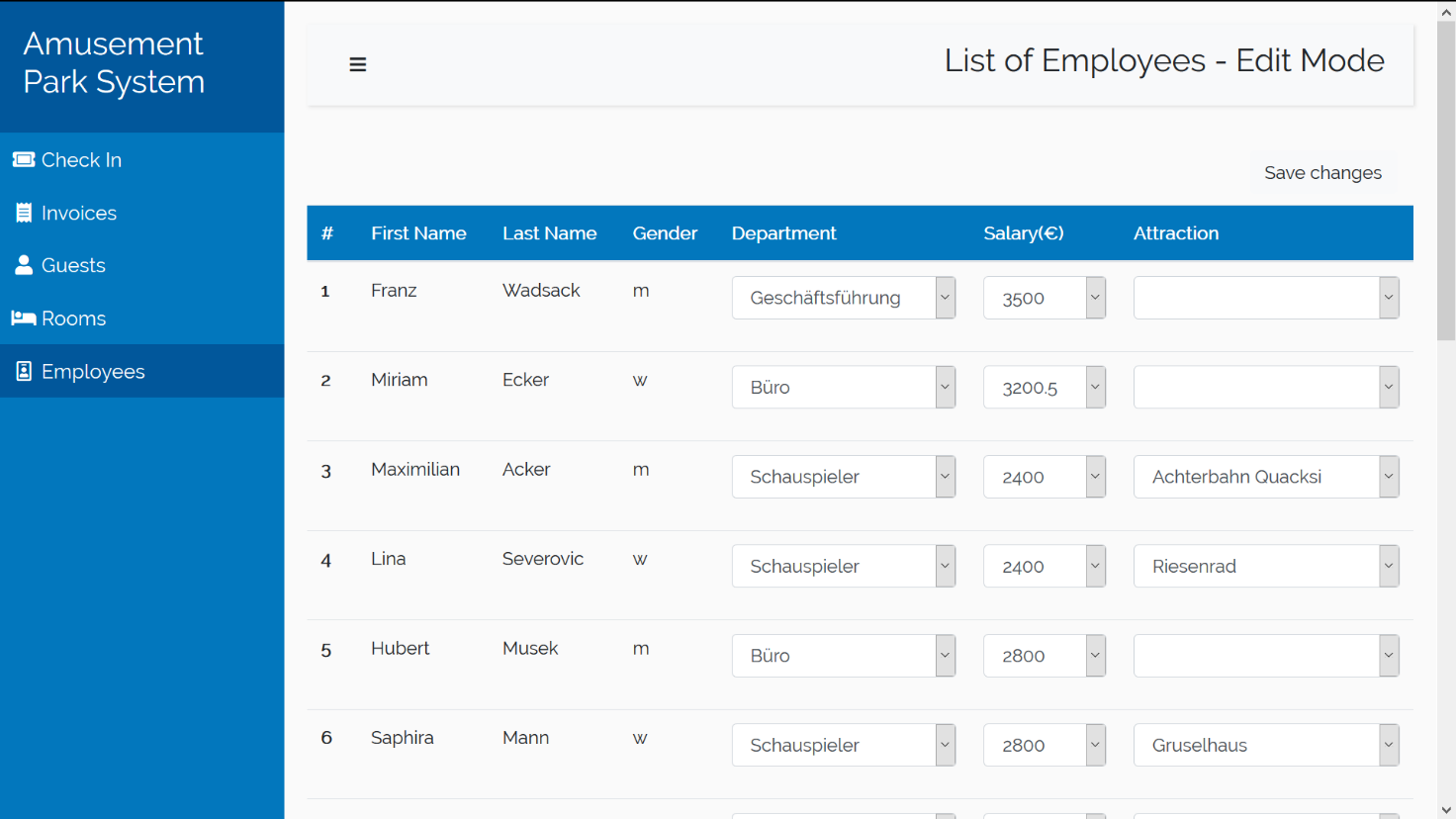


Abbildung 9: Personalverwaltung

1. Ermöglicht gaeste\_view\_grouped in Oracle 10g, siehe https://oracle-base.com/articles/misc/string-aggregation-techniques#collect [↑](#footnote-ref-1)