

## Online-Reservierungssystem für Ferienwohnungen

Eine Agentur, die Ferienwohnungen vermittelt, möchte ihr Angebot von Wohnungen im Internet zugänglich machen und Reservierungen online ermöglichen. Hierzu soll durch diese Übungen eine Datenbank-anwendung entwickelt werden. Erforderlich hierzu ist eine Datenbank, in welcher die notwendigen Daten abgelegt werden können.

In dieser Laborübung soll eine vollständige Datenbankapplikation entwickelt werden. Verschiedene Teilschritte müssen rechtzeitig zu den angegebenen Terminen abgegeben werden.

Eine erste Informationsbedarfsanalyse hat zu folgenden Ergebnissen geführt:

- **Ferienwohnungen** befinden sich in einem bestimmten **Land** und haben eine bestimmte Anzahl an **Zimmern**, einen **Namen**, eine **Größe in qm**, eine **Adresse** und einen **Preis pro gemieteten Tag**, angegeben in Euro. Für die Wohnungen sollen deren **Ausstattungen** gespeichert werden (z.B. Sauna, Schwimmbad, Tennisplatz, ...). Eine genaue Beschreibung der Ausstattungen (z.B. die Größe des Schwimmbads) ist nicht notwendig. Es sollen **mehrere Bilder** von der Ferienwohnung abgelegt werden.
- **Touristenattraktionen** (z.B. Freizeitparks, Badestrände, Sehenswürdigkeiten) werden mit **Name** und kurzer **Beschreibung** abgelegt. Für jede **Ferienwohnung** wird die **Entfernung** zu einigen nahegelegenen Touristenattraktionen gespeichert. Touristenattraktionen, die mehr als 50 Kilometer entfernt sind, sollen grundsätzlich nicht gespeichert werden.
- In der **Benutzungsoberfläche** können Kunden ihre gewünschte **Reisezeit**, **gewünschte Anzahl Zimmer**, **gewünschter Preis**, **gewünschtes Land** und optional **ein oder mehrere Ausstattungen** angeben. Daraufhin werden alle Ferienwohnungen angezeigt, die diese Eigenschaften besitzen und im gewählten Zeitraum frei sind. Dabei werden alle Eigenschaften der Ferienwohnung, ihre Bilder, Ausstattungen die durchschnittliche Kundenbewertung dargestellt.
- Kunden können Ferienwohnungen **online buchen**. Dazu müssen Sie sich zunächst mit ihrem **Namen**, **Mailadresse**, **Adresse** und **Bankverbindung (IBAN)** registrieren. Bei der Registrierung können sie entscheiden, ob sie den Newsletter abonnieren wollen. Über ihre **Mailadresse** und ein **Passwort** können sich Kunden später einloggen und müssen Ihre Daten nicht nochmals angeben. Die Korrektheit der Mailadresse soll durch einen regulären Ausdruck überprüft werden. Kunden sollen später nach dem Nachname sortiert werden können. Für eine Buchung müssen sie eine Ferienwohnung auswählen und den **Reservierungszeitraum** angeben. Der Mindestbuchungszeitraum ist drei Tage. Direkt nach der Buchung erhalten Kunden eine **Buchungsnummer**, welche das System automatisch vergibt. Das Datum der Buchung wird gespeichert.
- Einige Tage nach Buchung erhalten Kunden per Post eine **Rechnung mit einer Rechnungsnummer**, **Rechnungsdatum** und dem **Rechnungsbetrag**. In Rechnungen können ein oder mehrere **Anzahlungen** vereinbart werden. Jede durchgeführte Anzahlung wird mit **Geldbetrag** und **Zahlungsdatum** gespeichert. Nachdem die gesamte Rechnung bezahlt ist, ist die Buchung abgeschlossen.
- Nach dem Buchungszeitraum können Kunden eine Ferienwohnung durch eine **Anzahl Sterne** bewerten. Fünf Sterne steht für die beste Bewertung, ein Stern als schlechteste Bewertung. Es soll sichergestellt sein, dass nur Ferienwohnungen bewertet werden, die auch gebucht wurden. Das **Datum der Bewertung** soll automatisch gespeichert werden, die Bewertung darf erst nach dem Buchungszeitraum durchgeführt werden.
- Mitarbeiter der Agentur sollen alle in der Datenbank enthaltenen Daten pflegen können. Diese Pflege beinhaltet z.B. das Hinzufügen/Löschen einer neuen Ausstattung, Ferienwohnung oder Touristenattraktion.

### **Teil 1 ( Konzeptionelles Modell )**

Entwickeln Sie ein konzeptionelles Datenbank-Design unter Verwendung eines Entity Relationship-Modells. Das ER-Modell soll mit dem Tool JERM entwickelt werden. Betrachten Sie dafür auch die Anfragen aus Teil 4, die an das Modell gestellt werden sollen.

### **Teil 2 ( Relationenmodell )**

Entwickeln Sie aus dem ER-Modell ein Relationen-Modell in der Boyce-Codd-Normalform.

### **Teil 3 ( Spezifikation )**

Dokumentieren Sie Einschränkungen (Constraints), die sich aus der Analyse des Realweltausschnitts ergeben. Use Cases sollen durch ein UML-Use Case-Diagramm beschrieben werden. Transformieren Sie das erarbeitete ER-Modell in ein Relationen-Modell in Boyce-Codd-Normalform. Formulieren Sie im Rahmen eines Datenschutzkonzepts Zugriffsberechtigte und Zugriffsrechte. Auf der dritten Seite dieses Aufgabenblatts sind die abzugebenden Inhalte nochmals genau aufgezählt.

Als Ergebnis des ersten Teils des Praktikums muss ein Bericht abgegeben werden. Der Bericht soll die Ergebnisse der konzeptionellen Entwurfsphase dokumentieren und muss folgendes beinhalten:

- ER-Modell mit Kardinalitäten unter Verwendung des Tools JERM
- Relationen-Modell in Boyce-Codd Normalform
- Dokumentation aller Einschränkungen des Datenbankmodells, z.B. Datentyp, Unterscheidung Muss- und Kannattribute, Constraints, Optionen bei FOREIGN KEY (ON DELETE ...). Ein Beispielfile, wie diese Einschränkungen angegeben werden können, ist in Moodle vorhanden.
- UML Use-Case-Diagramm
- Zugriffsmatrix für alle Benutzer auf sämtliche Relationen des Modells
- Layout der Internet-Seiten zum Registrieren von Benutzern und Suchen von Ferienwohnung

Dieser Bericht soll als eine PDF-Datei abgegeben werden. Das ER-Modell muss bereits im Teil 1 zuvor abgegeben werden um Folgefehler zu vermeiden. Die Abgabe des Berichts erfolgt in Moodle. Der Abgabetermin ist fest vorgegeben, eine nachträgliche Abgabe ist nicht möglich.

### **Teil 4 ( SQL-Tabellen und Daten )**

Speichern Sie Daten für die Ferien-Onlinebuchung in Oracle 19c mit beispielhaften Daten. Es soll also folgendes gemacht werden. Es empfiehlt sich, die Daten so zu definieren, dass sie mit ihnen die Korrektheit der folgenden Anfragen überprüfen können.

1. Definieren Sie alle Tabellen Ihrer Spezifikation mit den spezifizierten Constraints.
2. Fügen Sie beispielhafte Daten in Ihre Datenbank ein. Es sollen genügend Daten in die Datenbank eingetragen werden, um die Funktionsweise der später folgenden SELECT-Anfragen zu verdeutlichen. Jede Anfrage sollte mindestens zwei Lösungen ergeben.
3. Vergeben Sie durch GRANT-Befehle einem freien dbsys-Account die Zugriffsrechte eines Kunden.
4. Überprüfen Sie Ihre Datenbank durch einfache SELECT-Anfragen. Die Anfragen sollen vom Kunden-Account aus durchgeführt werden.

### Teil 5 ( SQL-Suche )

Geben Sie für folgende Fragen SQL-Anfragen an. Alle Anfragen sollen unter dem Kunden-Account durchgeführt werden. Den letzten Befehl können Sie im letzten Aufgabenteil wiederverwenden.

- a) Wie viele Ferienwohnungen wurden noch nie gebucht?
- b) Welche Kunden haben eine Ferienwohnung bereits mehrmals gebucht?
- c) Welche Ferienwohnungen in Spanien haben durchschnittlich mehr als 4 Sterne erhalten?
- d) Welche Ferienwohnung hat die meisten Ausstattungen?
- e) Wie viele Reservierungen gibt es für die einzelnen Länder? Länder, in denen keine Reservierungen existieren, sollen mit der Anzahl 0 ebenfalls aufgeführt werden. Das Ergebnis soll nach der Anzahl Reservierungen absteigend sortiert werden.
- f) Wie viele Ferienwohnungen sind pro Stadtnamen gespeichert?
- g) Geben Sie für einen Kunden Empfehlungen aus. Wenn ein Kunde K1 eine Wohnung W1 mit 5 Sterne bewertet hat und ein Kunde K2 die Wohnung W1 ebenfalls mit 5 Sterne bewertet hat, dann sollen alle weiteren Wohnungen, die K2 mit 5 Sternen bewertet hat als Empfehlung für K1 ausgegeben werden.
- h) Untersuchen Sie, ob es Doppelbuchungen gibt. Geben Sie dazu alle Ferienwohnungen aus, für die es zwei Buchungen gibt, die die Ferienwohnung am gleichen Tag reservieren. Hinweis: es ist kein Problem, wenn ein Kunde die Ferienwohnung verlässt, und am gleichen Tag ein anderer Kunde einzieht.
- i) Welche Ferienwohnungen mit Sauna sind in Spanien in der Zeit vom 1.11.2021 – 21.11.2021 noch frei? Geben Sie den Ferienwohnungs-Namen und deren durchschnittliche Bewertung an. Ferienwohnungen mit guten Bewertungen sollen zuerst angezeigt werden. Ferienwohnungen ohne Bewertungen sollen am Ende ausgegeben werden.

### Teil 6 ( SQL-Übung )

a)

Buchungen sollen nicht gelöscht werden, sondern nur „storniert“. Erweitern Sie ihr Datenbankschema um eine Tabelle für diese „stornierte Buchungen“. Nutzen Sie das Triggerkonzept, um sicherzustellen, dass beim Löschen einer Buchung ein passender Eintrag in der Tabelle für „stornierte Buchungen“ vorgenommen wird. Für stornierte Buchungen soll zusätzlich das Stornierungsdatum automatisch hinzugefügt werden.

b)

Damit ein Mitarbeiter der Agentur schnell erkennen kann, ob ein Kunde ein geschätzter Stammkunde ist, sollen Informationen wie die Anzahl bisher getätigter Buchungen und die Summe aller bisher gezahlter Rechnungsbeträge zusammengefasst werden. Da diese Informationen sehr verteilt in der Datenbank gespeichert sind, soll ein View erstellt werden, der diese Informationen zusammenfasst. Definieren Sie einen View Kundenstatistik, der folgende Attribute besitzt: Kundennummer, Anzahl Buchungen, Anzahl Stornierungen, Summe aller Zahlungen. Falls ein Kunde noch keine Buchungen bzw. keine Stornierungen durchgeführt hat, soll die Zahl 0 erscheinen (nicht NULL).

c)

Angenommen, der Befehl aus Aufgabe 4i ist zu langsam. Definieren Sie Indexe, die die Suche beschleunigen.

## Teil 7 ( Programmierung des Systems )

Implementieren Sie mit Hilfe von eingebetteten SQL-Statements ein Java-Programm, das folgende Funktionen besitzt. Das Programm soll auf die Datenbank durch einen Kundenaccount zugreifen, der die für Kunden spezifizierte Zugriffsrechte besitzt. Ein Kunde soll folgende beiden Operationen durchführen können:

1. Suche nach Ferienwohnungen:  
Ein Land, ein Anreise- und ein Abreisetermin sollen von der Tastatur eingelesen werden. Optional soll der Kunde auch eine Ausstattung angeben können, die die gesuchte Ferienwohnung haben soll. Anschliessend sollen alle Ferienwohnungen ausgegeben werden, die die angegebenen Eigenschaften erfüllen, zusammen mit ihrer durchschnittlichen Bewertung. Sie können dabei den letzten SQL-Suchbefehl von Teil 4 wiederverwenden.
2. Buchen von Ferienwohnungen:  
Eine Ferienwohnung soll für einen Kunden über einen bestimmten Zeitraum gebucht werden. Dazu muss aus dem Suchmenü eine Ferienwohnung ausgewählt werden können. Die Buchungsnummer soll das System automatisch vergeben.

Entwickeln Sie eine grafische Oberfläche, in denen die beiden oberen Use Cases umbesetzt sind. Die Auswahl von Länder und Ausstattungen soll durch eine Dropdown-Liste umgesetzt werden, in der alle Länder bzw. Ausstattungen aufgelistet werden.

Hinweis: Schliessen Sie Ihre Transaktionen mit Commit bzw. im Fehlerfall mit Rollback ab.

## Hinweise für die Durchführung des Labors

Die Vergabe von Zugangsberechtigungen für ORACLE19c erfolgt zu Beginn des Praktikums. Für die Durchführung des Praktikums werden die Tools JERM 2.1 und SQL Developer verwendet. Für den Praktikumsschein müssen alle Teile gelöst werden.

In den Übungen sind 6 Abgaben vorgesehen. Bis auf Teil 2 finden alle Abgaben im Poolraum durch Vorzeigen und Erklären der Lösung statt. Die Lösungen für Teil 3-6 müssen unter Oracle19c und den im Poolraum stehenden Rechnern vorgeführt und erklärt werden können. Eine Abgabe per Mail ist nicht möglich.

Die Abgabetermine sind im Terminplan in Moodle angegeben.

Anmelden bei Oracle-Server:

- Hostname, Port und SID eintragen
- Verbindungsname mit einem beliebigen Name benennen
- Eigener Benutzername (dbsys-Account) eingeben, Passwort ist gleich wie Benutzername

Es empfiehlt sich das eigene Passwort zu ändern, damit es nicht zu Verwechslungen kommen kann.

Datenbankverbindung erstellen/wählen

Verbindungsna...	Verbindungsdet...
dbsys39	dbsys39@//ora...
eck	eck@//oracle12...
eck19c	eck@//oracle19...

Name:

Datenbanktyp:

**Benutzerinformationen** Proxybenutzer

Authentifizierungstyp:

Benutzername:  Rolle:

Kennwort:  ☒ Kennwort speichern

Verbindungstyp:

**Details** Erweitert

Hostname:

Port:

☒ SID:

☐ Service-Name:

Status:

Nach Anmelden Passwort ändern

- alter user <username> identified by <newpassword>;

Gelegentlich sollte der Oracle "Recycle Bin" gelehrt werden

- purge recyclebin;