

1. Umwandlung Klassenmodell -> Datenmodell

- 1. Auswahl der persistenten Klassen (Entity Klassen)
- 2. Jede persistente Klassen bekommt ein OID
- 3. Abbildung einer Klasse auf eine Entität
 - a. OID wird Schlüsselattribut
 - b. Jedes persistente Attribut wird auf Attribut der Entität abgebildet (bei Person wird OID zu P_ID)
- 4. Abbildung von Beziehungen
 - Beziehungen ins logische Datenmodell abbilden
- 5. Überführung ins physikalische Modell
 - jedes Entity wird auf eigene Tabelle mit Primärschlüssel abgebildet
 - jedes Attribut bekommt einen Datentyp und wird in der jeweiligen Spalte abgebildet
 - 1:N Beziehungen werden auf Fremdschlüssel und Detailtabellen abgebildet
- Vertikale Partitionierung
- Horizontale Partitionierung

2. Erklärung von Transaktionen

- Mehrere SQL Befehle müssen als Einheit ausgeführt werden!
- Eine Transaktion ist eine Menge von DB-Operationen, die eine DB von einem konsistenten Zustand in einen weiteren konsistenten Zustand überführt!
- Entweder alle Transaktionen oder gar keine!!!

3. wann/warum aktive dbms? Was versteht man unter aktiven DBMS?

- Aktives DBMS ist, wenn durch eine interne oder externe Aktion(en) auf interne oder externe Ereignisse reagiert wird. Ereignisse werden durch Trigger beschrieben
Wann? Sobald auf ein Ereignis reagiert werden soll!
 - Steuerung von Abläufen: Benachrichtigungen bei Fehlern, Überwachung von Ausnahmeregelungen

4. Erklärung: aktives Datenbanksystem

Abschätzung: Du möchtest ein Softwareprojekt mit persistenten Teilen schreiben -> möchtest Du dich da um die Implementierung der Persistenz selber kümmern (-> statisches SQL -> z.B. SQLJ) oder lieber die Speicherung an einen O/R-Mapper abgeben (z.B. Hibernate)...unter Umständen dann noch auf die Flexibilität etc. eingehen?

5. Trigger schreiben

- CREATE OR REPLACE TRIGGER gehalt_ausgeben
Before delete or insert or update on Angestellte
For each row
When (new.angestelltennr > 0)
Declare
 Diff NUMBER;
Begin
 Diff := :new.gehalt - :old.gehalt
END;

6. Was sind Stored Procedures?

- Sind allein stehende Einheiten, die Global definiert und verfügbar sind
- Werden in der Datenbank abgelegt einmal!
- Prozedur (ohne Rückgabetyt) oder Funktion (mit Rückgabebefehl)
- DECLARE -> AS -> Begin -> Exception -> End

7. Nennen Sie die Neuerungen in SQL:2003.

- Objekt relationale Erweiterungen
- Large Objects (LOBs -> BLOB, CLOB, NLOB)
- Benutzerdefinierte Datentypen (UDTs)
- Typkonstruktoren
- Collections
- Trigger
- XML
- Rekursive Anfragen
- Hierarchien von Tabellen und Sichten
- Benutzerdefinierte Methoden, Funktionen, Prozeduren

8. Ich möchte einen Film/Fotos in einer Datenbank ablegen, was für Datentypen stehen mir in Oracle ab Version 9.0 zur Verfügung?

- Kann in LOBs gespeichert werden! Large Object bzw als Typ BLOB, binary large object
- Dieses wird bei Oracle wahrscheinlich als BFILE gespeichert, also extern, nur mit Referenz, die das OS verwaltet

9. Zeilen und Spaltentypen?

- Strukturierter Typ, selber Definiert
- Spaltentyp:
 - Typen der Attribute z.B. Bilder, Text, Video
 - Verbesserung der Verwaltung von Multimedia Objekten
- Zeilentyp
 - Modellierung von Objekten mit Beziehungen und Verhalten
 - Z.B. Person, Angestellter, Student
 - Verbesserung der Verwaltung von Business Objekten
 - Nach P_ID

10. Referenztypen

- ein strukturierter Typ hat einen dazugehörigen Referenztyp
- Referenztyp kann wie ein normaler Typ benutzt werden
- Benutzer Generiert, System generiert, Abgeleitet von einer Menge von Attributen

11. Was sind Multimengen? Unterschied zu Array?

- Eine Multimenge ist eine ungeordnete, unbeschränkte Kollektion von Werten
- Multimengen werden im Vergleich zu Arrays mit einer Vielzahl an typspezifischer Funktionen und Prädikaten bereitgestellt:
 - MULTISSET()
 - Überprüfungen, ob eine Menge vorhanden ist

12. Was sind varrays und wie werden sie angelegt?

- Geordnete Menge von Elementen, alle vom gleichen Datentyp
- Jedes Element hat einen Index, Position in Array

- Größe des Arrays = Anzahl von Elementen
- Deklaration: Maximale Anzahl von Elementen
- Löschen der Elemente durch Null setzen

13. Erklären sie Nested Tables?

- Geschachtelte Tabellen (Tabelle in Tabelle)
- Ungeordnet
- Keine Feste Obergrenze an Daten

14. Schreiben sie Befehle auf mit denen zwei Benutzerdefinierte Typen erzeugt werden und den Befehl für die Erzeugung der Tabellen inkl Nested Table.

- 1. Typ bauen:
*CREATE TYPE **planTyp** AS OBJECT*
*(**Plan_ID** NUMBER, **Nummer_Figuren** NUMBER,*
***ARCHTEKTEN** Nametyp)*
- 2. Tabelle erstellen
*CREATE TABLE **TabellePlan** of *planType* (PRIMARY KEY*
(Plan_ID)
- 3. Typ für innen Tabelle definieren, der als Typ planTyp hat!!
*CREATE OR REPLACE TYPE **Nt_Plaene** AS TABLE OF REF*
Plantype
- 4. Type für die innen tabelle erstellen
*CREATE TYPE **ProjektTyp** AS OBJECT*
*(**ProjektID** NUMBER, **NAME_P** Varchar2(30), **hat_Plan***
Nt_Plaene)
- 5. Tabelle erstellen!
*CREATE TABLE **TabelleProjekte** of *ProjektTyp**
*(**Primary Key** (Projekt_ID), **UNIQUE** (Name_P), *NESTED TABLE**
*hat_plan **STORE AS** PlaeneListe*

15. Hibernate Mappingfile

16.SQLJ - Hibernate vergleichen

17.cmt und bmt

18.jpa erklären

19.Java Persistence API erklären (beide geschoben😄)

20.OIDs in ORDBMS (manuell, automatisch, ...)

21.

22. Welche Typen an EJBs gibt es?Erklärung

23.Wozu dienen Remot und Local Interfaces?

24.Vergleichen und bewerten Sie EJBs und SQLJ.

25.Transaktionsmanagement in Entity Beans (Stichwort required)

26. Erläutern Sie die Unterschiede zwischen EJB 2.0 und EJB 3.0.

27.Welche Zustände gibt es bei Session Beans?