

Aufgabe 2 Datenbankentwurf

Die Unternehmensberatung Besser-Schneller-Billiger GmbH legt Ihrer Firma den folgenden Datenbankentwurf für die Projektverwaltung vor:

Alle Daten werden in einer Tabelle **Pers_Proj** gehalten ("schneller"!):

Pers#, Abt# und Proj# bezeichnen Personal-, Abteilungs- und Projektnummern, und PersName, AbtName und ProjName die entsprechenden Namen. Der Projektname bezeichnet den Beginn des Projekts in der Form 'Pmmyyyy'. ProjTage enthält die Anzahl der T, die die MitarbeiterInnen in dem jeweiligen Projekt abgeleistet haben.

Pers_Proj	Pers#	PersName	Abt#	AbtName	Proj#	ProjName	ProjTage
	101	Melchior	1	Motoren	11,12	P012005, P032006	60,40
	102	Schmid	2	Karosserie	13	P012006	100
	104	Seeler	2	Karosserie	11,12,14	P012005, P032006, P012007	20,30,40
	105	Rabe	1	Motoren	13,14	P012006, P012007	50,30
	107	Meister	5	Marketing	10,12	P072004, P032006	40,40
	109	Scholz	5	Marketing	10	P072004	90
	201	Adamski	1	Motoren	12	P032006	80
	222	Berger	3	Rechenzentrum	15	P122006	120
	225	Waltemate	3	Rechenzentrum	12,15	P032006, P122006	10

a) Erstellen Sie in Ihrem Oracle-Schema diese Tabelle als Projektdatenbank. Bei auftretenden Schwierigkeiten wie dem Fehlen eines benötigten Datentyps ändern Sie das Tabellenschema entsprechend.

```
create table Pers_Proj
(Pers# NUMBER(3) NOT NULL,
 PersName VARCHAR2(20) NOT NULL,
 Abt# NUMBER(1) NOT NULL,
 AbtName VARCHAR(20) NOT NULL,
 Proj# VARCHAR(20),
 ProjName VARCHAR(20),
 ProjTage VARCHAR(20));
```

Der Entwurf ist nicht praxisrelevant, weil Projekte, die im Verlauf geändert werden sollen, nur schwer zu ändern sind. Auch das Heraussuchen wie beispielsweise von Kandidaten an einem Projekt ist mit diesem Entwurf nur sehr umständlich möglich.

Überlegung 1: Array

Weil in einigen Datensätzen Eigenschaften mehrere Merkmale enthalten sein können (Proj#, ProjName, ProjTage), empfiehlt sich die Hinzunahme von Arrays. Der Index steht beispielsweise für eine fortlaufende Nummerierung der Projekte, an denen die

Person, der der Datensatz zugeordnet ist, beteiligt ist. Von Vorteil ist das schnelle Auffinden von Projektdaten zu einer Person. Schlecht ist, wenn man sich alle Projektdaten auflisten lassen will (z.B. eine Tabelle mit allen Projektnamen oder die Summe aller Projektstage in einem Projekt), weil dies nur umständlich möglich und mit einem hohen Rechenaufwand verbunden ist.

Überlegung 2: Verteilung auf mehrere Tabellen

Der Vorteil bei einer Verteilung auf mehrere Tabellen bringt den Effekt, daß die Redundanz schwächer ausfällt und somit der Speicherverbrauch merklich sinkt. Der ProjName zu jedem Proj# muss nur einmal vermerkt werden. Von Nachteil ist die sinkende Konsistenz (passt eine Veränderung zu jedem verknüpften Datensatz?) und der höhere Verarbeitungseffekt, wenn für eine Abfrage auf mehrere Tabellen zugegriffen werden muss.

Überlegung 3: Kombination

Um die Vorteile aus beiden vorgehenden Überlegungen möglichst effektiv nutzen zu können, bietet sich eine situationsbedingte Kombination an. Zunächst werden die AbtNamen betrachtet. Hier fallen die speicherintensiv ausfallenden Bezeichnungen („Karosserie“, „Marketing“, „Rechenzentrum“) auf. Die Wiederholung dieser Namen und somit eine prozentual stark ausfallende Redundanz kann verhindert werden, indem eine neue Tabelle angelegt wird, deren Primärschlüssel Abt# ist. In dieser Tabelle werden nur die AbtNamen abgelegt. Eine deutliche Verbesserung des Entwurfes stellt die Auslagerung der Kombination Proj# mit ProjName dar. Auch die Redundanz, aber viel mehr die Funktionalität sprechen dafür, denn mit der aktuellen Lösung kann man nur über Umwege zu einem Projekt mit vorgegebenem Namen die zugehörigen Personen finden. In der Tabelle Pers_Proj bleibt nur ein Array mit Fremdschlüssel Proj# und ein Array mit ProjTage zurück.

Überlegung 4: Statt Array nur Tabellen

Um kein Array verwenden zu müssen, bietet sich die Auslagerung in eine weitere Tabelle an. Diese Tabelle benötigt keinen Primärschlüssel, denn sie umfasst lediglich das Verhältnis Mitarbeiter – Projekte (und Tage). Somit wird diese Tabelle aus Pers#, Proj# und ProjTage bestehen und löst das Problem, daß Arrays aus der Personaltabelle durchgewählt werden müssten.

b) Welche Spalten oder Spaltenmengen sind Schlüsselkandidaten (warum?), welche wählen Sie als Primärschlüssel aus? Fügen Sie die entsprechenden Primärschlüssel-Constraints ein.

```
create table Pers
(Pers# NUMBER(3) NOT NULL,                                     Primärschlüssel
 PersName VARCHAR2(20) NOT NULL,
 Abt# NUMBER(1) NOT NULL,
 constraint Pers_Primary_Key primary key (Pers#));
```

```
create table Abt
(Abt# NUMBER(1) NOT NULL,                                     Primärschlüssel
 AbtName VARCHAR2(20) NOT NULL,
 constraint Abt_Primary_Key primary key (Abt#));
```

```
create table Projekte
(Proj# NUMBER(2) NOT NULL,                                     Primärschlüssel
 ProjName VARCHAR2(20),
```

```

constraint Proj_Primary_Key primary key (Proj#));

create table Pers_Proj
(Pers# NUMBER(3) NOT NULL,                               Fremdschlüssel
 Proj# NUMBER(2) NOT NULL,                               Fremdschlüssel
 ProjTage number(3) NOT NULL,
 constraint pers_vorhanden check
 (select Pers# from Pers where Pers.Pers# = Pers#),
 constraint proj_vorhanden check
 (select Proj# from Projekte where Projekte.Proj# = Proj#));

```

Das Einfügen der vorliegenden Daten muss wie folgt erfolgen:

```

insert into pers values (101, 'Melchior', 1);
insert into pers values (102, 'Schmid', 2);
insert into pers values (104, 'Seeler', 2);
insert into pers values (105, 'Rabe', 1);
insert into pers values (107, 'Meister', 5);
insert into pers values (109, 'Scholz', 5);
insert into pers values (201, 'Adamski', 1);
insert into pers values (222, 'Berger', 3);
insert into pers values (225, 'Waltemate', 3);

```

```

insert into abt values (1, 'Motoren');
insert into abt values (2, 'Karosserie');
insert into abt values (3, 'Rechenzentrum');
insert into abt values (5, 'Marketing');

```

```

insert into projekte values (10, 'P072004');
insert into projekte values (11, 'P012005');
insert into projekte values (12, 'P032006');
insert into projekte values (13, 'P012006');
insert into projekte values (14, 'P012007');
insert into projekte values (15, 'P122006');

```

```

insert into pers_proj values (101, 11, 60);
insert into pers_proj values (101, 12, 40);
insert into pers_proj values (102, 13, 100);
insert into pers_proj values (104, 11, 20);
insert into pers_proj values (104, 12, 30);
insert into pers_proj values (104, 14, 40);
insert into pers_proj values (105, 13, 50);
insert into pers_proj values (105, 14, 30);
insert into pers_proj values (107, 10, 40);
insert into pers_proj values (107, 12, 40);
insert into pers_proj values (109, 10, 90);
insert into pers_proj values (201, 12, 80);
insert into pers_proj values (222, 15, 120);
insert into pers_proj values (225, 12, 10);
insert into pers_proj values (225, 15, 10);

```

c) Welche Integritätsbedingungen erscheinen Ihnen sonst noch sinnvoll?

Es fehlen Trigger und Constraints, die die Manipulationen zwischen Pers_Proj und den Tabellen Pers und Abt vereinen.

d) Fügen Sie je eine neue MitarbeiterIn, eine neue Abteilung und ein neues Projekt ein. Schwierigkeiten? Protokollieren Sie sie!

Mitarbeiter: - die Abteilung muss bekannt sein

Abteilung: - keine Probleme

Projekt: - keine Probleme

e) Der Name der Abteilung 1 ändert sich – sie heisst ab sofort *Motorentchnik*. Führen Sie entsprechende Updates durch.

```
update abt set AbtName='Motorentchnik' where Abt#=1
```

f) Denken Sie sich drei Anfragen an Ihre Datenbank aus und stellen Sie sie.

g) Frau Berger und Herr Waltemate verlassen die Firma. Was passiert beim Löschen ihrer Einträge?

Es müssen nicht nur ihre Einträge in *Pers* gelöscht oder als gelöscht markiert werden, es müssen auch ihre Einträge in *Projekte* angepasst werden. Meine Empfehlung ist ein Löschbit, das gesetzt wird, und ein constraint, das verhindern soll, daß gelöschte Personen an einem Projekt beteiligt sind.

h) Erstellen Sie ein neues „gutes“ ER-Diagramm, entweder mit Papier und Bleistift oder mit dem PowerDesigner.

