SQL-Befehle für Schemaentwurf

(siehe dazu auch die entsprechenden Seiten im Skript)

Dieses Beiblatt beschreibt die Syntax der ORACLE SQL-Befehle zum Erzeugen von Tabellen, Indizes und Clustern. Es werden hier nur diejenigen Parameter erklärt, die Sie im Rahmen Ihres Entwurfes bestimmen sollten.

Weitere Informationen finden Sie im *Oracle7 Server SQL Reference Manual* unter *file:/home/stud/kraiss/oradoc/Welcome.html*

Erzeugen von Clustern

ORACLE bietet die Möglichkeit zum Erzeugen von Cluster-Indizes und Hash-Clustern. Tupel mit gleichen Werten im Clusterschlüssel bzw. mit gleichem Hash-Wert werden in Clustern möglichst in einem Block abgespeichert. Ein Cluster wird durch folgenden Befehl erzeugt:

CREATE CLUSTER *clustername* (attr1 datentyp1 [,attr2 datentyp2] ...)
[**PCTUSED** integer] [**PCTFREE** integer]
[**SIZE** integer [**K**|**M**]]
[**INDEX** | [**HASH IS** *attr*] **HASHKEYS** integer]

Die Attribute attr1, attr2,... geben den Clusterschlüssel an. Die Parameter PCTUSED und PCTFREE steuern den Füllgrad von Datenblöcken. Der Parameter **PCTUSED** bezeichnet dabei den Anteil (Angabe in Prozent) eines Datenblocks, der mindestens belegt sein sollte. Wird dieser Anteil unterschritten, so dürfen wieder Tupel in den Clusterblock eingefügt werden. **PCTFREE** gibt an, wieviel Prozent des Datenblocks mindestens frei sein sollten. Wird dieser Wert unterschritten, so dürfen in den Datenblock keine weiteren Tupel eingefügt werden, bis der minimale Füllgrad (PCTUSED %) wieder erreicht ist. Die Summe aus PCTUSED und PCTFREE darf den Wert 100 nie überschreiten. Der Parameter **SIZE** bestimmt, wieviele Bytes innerhalb eines Datenblockes für denselben Clusterschlüssel-Wert bzw. Hash-Wert verwendet werden. Der Wert darf die Größe eines Datenblockes (in unserem Fall 2048) nicht überschreiten. Wird die Angabe von SIZE weggelassen, so reserviert ORACLE einen Datenblock pro Schlüsselwert. Die Angabe von SIZE erfolgt in Bytes, wobei **K** und **M** für 'Kilobytes' bzw. 'Megabytes' stehen. Die Angabe von **INDEX** erzeugt einen Index-Cluster, während durch **HASHKEYS *n*** ein Hash-Cluster mit n verschiedenen Hash-Werten (Buckets) erzeugt wird. Der Wert n wird zuvor intern auf die nächst größere Primzahl aufgerundet. Der Default-Cluster ist INDEX. Durch **HASH IS *attr*** kann bei einem Hash-Cluster das Attribut *attr* direkt als Hash-Funktion verwendet werden. Es muß jedoch vom Typ NUMBER sein, muß äquivalent zum Cluster-Schlüssel sein und darf nur nicht-negative Werte enthalten. Wird diese Angabe weggelassen, so verwendet ORACLE eine Standard Hash-Funktion. Der Erzeugung eines Index-Clusters muß der Aufbau des entsprechenden Index folgen, damit auf die Daten zugegriffen werden kann. Die Tabellen, die in einem Cluster abgelegt werden sollen, müssen ebenfalls *nach* dem Cluster erzeugt werden.

Erzeugen von Indizes

Indizes werden durch folgende Anweisung erzeugt:

CREATE INDEX indexname **ON** { tabellenname(attr1 [,attr2]...)
| **CLUSTER** clustername }
[**PCTFREE** integer]

Durch die Angabe eines Tabellennamens und der zu indizierenden Attribute der Tabelle wird ein mittelbarer Index auf der entsprechenden Relation aufgebaut. Durch Angabe von **ASC** bzw. **DESC** wird das Attribut innerhalb des Index aufsteigend bzw. absteigend sortiert. Wird statt eines Tabellennamens **CLUSTER *clustername*** angegeben, so wird ein Cluster-Index über dem Cluster mit der Bezeichnung *clustername* aufgebaut. Der Cluster *clustername* muß als Index-Cluster bereits erzeugt worden sein. Der Wert von **PCTFREE** bestimmt, wieviel Prozent der Index-Datenblöcke beim Aufbau des Index mindestens unbelegt bleiben.

Erzeugen von Tabellen

Die Tabellen, die Sie in Ihrem Entwurf verwenden, sind bereits in ihrem Grundgerüst durch das konzeptionelle Schema gegeben. Die grundlegenden Tabellen-Definitionen finden Sie im CIP-Pool in der Datei **~kraiss/P4/Grundschema.sql**. Sie dürfen keine der hier bereits spezifizierten Integritätsbedingungen entfernen, auch wenn dadurch die Performance verbessert werden würde (das ist eben der Preis, den man für Aufrechterhaltung der Integrität zahlen muß). Sie dürfen jedoch die im folgenden Befehlsaufbau beschriebenen Parameter zur Speicherung der Relationen ergänzen.

CREATE TABLE tabellenname(attr1 datentyp1 [constraints]
[, attr2 datentyp2 [column_constraint]]...)
[[**PCTFREE** integer] [**PCTUSED** integer] | [**CLUSTER** clustername (attr1 [, attr2]...)]]

Als Constraints wollen wir nur diejenigen vom Typ **NOT NULL, PRIMARY KEY und REFERENCES Relation(attr)** betrachten, deren Semantik dem Skript entnommen werden können. Wichtig ist hierbei, daß durch Angabe von **PRIMARY KEY** intern ein Index über dem Primärschlüssel aufgebaut wird, der auch vom Optimierer verwendet wird.
Soll die Tabelle in einem Cluster abgelegt werden, so muß durch **CLUSTER clustername** der die Tabelle enthaltende Cluster angegeben werden, zusammen mit den Attributen (attr1, attr2...) der Tabelle, die dem Clusterschlüssel entsprechen. Es muß hier beachtet werden, daß das an einer bestimmten Position angegebene Tabellenattribut im Typ identisch ist zum Clusterschlüssel-Attribut an derselben Position. Wird die Tabelle nicht in einem Cluster abgelegt, so wird durch PCTUSED festgelegt, wieviel Prozent der Datenblöcke minimal belegt sein sollten. Fällt der Füllgrad unter diesen Wert, so können wieder Tupel in den Datenblock eingefügt werden. Durch PCTFREE wird vorgegeben, wieviel Prozent des Datenblockes mindestens freigehalten werden sollen. Wird der Wert unterschritten, so dürfen solange keine Tupel in den Block eingefügt werden, bis weniger als *PCTUSED* Prozent des Blockes belegt sind. Die Summe aus beiden Angaben darf 100 nicht überschreiten. Der Default-Wert von PCTFREE ist 10. von PCTUSED 40.

Ermitteln des Query-Plans

Sie können den Query-Plan des ORACLE Optimierers, wie in der dritten praktischen Übung, durch Aufruf des Skriptes **~kraiss/P4/sp.sql** ermitteln (*sp* für 'show plan'). Durch Aufruf von *@sp 'SELECT * FROM KUNDEN WHERE KundenNr=123'* wird beispielsweise der Query-Plan für Query 1.1 der Anwendungsmodule ermittelt.