

Aufgabe 9 feste und variable Satz­längen

Betrachten Sie die Tabelle employees des Schemas HR. Nehmen Sie eine Seitengröße von 512 Bytes und (als Vereinfachung) Seitengröße = nutzbarer Speicherbereich an .

Oracle speichert numerische Daten in einem Format mit variabler Länge mit einem Byte für den Exponenten und maximal 20 Bytes für die Mantisse. Number(p) erfordert $\lceil (p+s)/2 \rceil + 1$ Bytes mit s=0 für positive und s=1 für neg. Zahlen. Der Maximalplatz beträgt 21 Bytes.

Das interne Datenformat für Datumsangaben umfasst 7 Bytes (feste Länge)

Name	Null?	Typ
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
EMAIL	NOT NULL	VARCHAR2(25)
PHONE_NUMBER		VARCHAR2(20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
SALARY		NUMBER(8,2)
COMMISSION_PCT		NUMBER(2,2)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)

a) Nehmen Sie an, jedes Attribut wäre mit einem Datentyp fester (also maximaler) Länge vereinbart. Berechnen Sie die benötigten Satz­längen.

EMPLOYEE_ID:	NUMBER(6)	5
FIRST_NAME:	VARCHAR2(20)	20
LAST_NAME:	VARCHAR2(25)	25
EMAIL:	VARCHAR2(25)	25
PHONE_NUMBER:	VARCHAR2(20)	20
HIRE_DATE:	DATE	7
JOB_ID	VARCHAR2(10)	10
SALARY	NUMBER(8,2)	6
COMMISSION_PCT	NUMBER(2,2)	3
MANAGER_ID	NUMBER(6)	5
DEPARTMENT_ID	NUMBER(4)	4

Summe 130 Byte

b) Berechnen Sie Unter- oder Obergrenzen der benötigten Satz­längen bei variabler Satz­länge. Wie sind die Speicherauslastung und der Speicherbedarf bei 100000 Datensätzen (fester Länge), wenn man keine Spannsätze zulässt?

Hinweis: Sie können sich anhand der Oracle Dokumentation näher informieren – für diesen Bereich bspw. unter Oracle concepts, Built-in datatypes.

Die Obergrenze ist, wie in a) angegeben, 130 Byte.

Untergrenze:

EMPLOYEE_ID:	NUMBER(6)	5
FIRST_NAME:	VARCHAR2(20)	1
LAST_NAME:	VARCHAR2(25)	1
EMAIL:	VARCHAR2(25)	1
PHONE_NUMBER:	VARCHAR2(20)	1
HIRE_DATE:	DATE	7
JOB_ID	VARCHAR2(10)	1
SALARY	NUMBER(8,2)	6

COMMISSION_PCT	NUMBER(2,2)	0
MANAGER_ID	NUMBER(6)	0
DEPARTMENT_ID	NUMBER(4)	0
Summe		23 Byte

Aufgabe 10 Datensätze in Blöcken

2000000 Datensätze der Länge 200 Bytes sollen in Blöcken der Größe 4KByte untergebracht werden. Berechnen Sie den Blockungsfaktor sowie die Anzahl der Blockzugriffe einer erfolgreichen und einer erfolglosen Suche nach einem Datensatz. Wie lange dauern die Suchen unter der Annahme von 10 ms pro Blockzugriff und 30 ns als Durchschnittszeit für die Verarbeitung eines Datensatzes?

Umrechnung der Blockgröße: $4 \text{ KByte} = 4 * 1024 = 4096 \text{ Byte}$
 $4096 / 200 = 20$ vollständige Datensätze pro Block
 $2.000.000 / 20 = 100.000$ Blöcke

Durchschnittlicher Zugriff: $100.000 \text{ Blöcke} / 2 = 50.000 \text{ Blöcke}$,
 Zugriffszeit: $10 \text{ ms} = 10.000.000 \text{ ns}$, $10.000.000 + (30 * 20) \text{ ns} = 10.000.600 \text{ ns}$

$50.000 * 10.000.600 = 500.030 \text{ ms} = \sim 500 \text{ Sekunden} = 8,3 \text{ Minuten}$

Aufgabe 11 Pufferverwaltung

a) Beschreiben Sie die Wirkungsweise der in der Vorlesung beschriebenen Ersetzungsstrategien der Pufferverwaltung (Folie 79ff.) anhand des ebenfalls in der Vorlesung beschriebenen Beispiels des (natürlichen) Verbundes zweier Tabellen unter Einsatz des nested – loop – joins (Folie 82).

FIFO-Strategie (was zuerst reingeht, wird zuerst entnommen)
 Jede Seite muss wenigstens einmal geladen werden. Wenn der Puffer voll ist, wird die Seite, welche am längsten im Puffer steht, gelöscht. Weitere Kriterien gibt es nicht.
 Vorteile: „Rückwärts-Navigation“ ist sehr schnell
 Nachteile: Stark frequentierte Seiten müssen periodisch neu geladen werden

LFU-Strategie (am seltensten referenzierte Seite wird ersetzt)
 Hier spielt nicht das „Alter“ eine Rolle, sondern die Anzahl der Zugriffe. Kann von Vorteil sein, aber jüngere Seiten haben schlechtere Chancen, eine Prioritätsstellung im Puffer zu bekommen als ältere Seiten.
 Vorteile: Speicher muss nur bei häufiger Verwendung von neuen Seiten öfter ausgetauscht werden
 Nachteile: Neue Seiten immer alten Seiten gegenüber benachteiligt

LRU-Strategie (am längsten nicht mehr referenzierte Seite wird ersetzt)
 Es werden Seiten im Puffer ausgetauscht, die am längsten nicht mehr referenziert wurden. Hier wird also ein Datestemp gesetzt, das für Relevanz sorgt.
 Vorteile: Laufend verwendete Seiten bleiben im Puffer
 Nachteile: Neue und lange nicht mehr aktivierte Seiten müssen nachgeladen werden

CLOCK-Strategie (am letzten häufig referenzierte Seite wird ersetzt)
 Seiten, die in der jüngsten Zeit am seltensten referenziert wurden, werden ersetzt. Zwei Kriterien kommen hier zum Tragen. Es werden also die LFU- und LRU-Strategien vereinigt.
 Vorteile: Praxisrelevante Seiten bleiben lange im Speicher, Nachteile der anderen Strategien werden weitestgehend ausgeräumt.

b) Wie sieht im Vergleich dazu der Ablauf aus, wenn die MRU (most recently used) –

Strategie eingesetzt wird, die diejenige Seite im Puffer ersetzt, die zuletzt referenziert wurde?

Betrachten Sie auch die Fälle, dass die Anzahl der Seiten der Tabelle B kleiner oder viel größer ist als die Anzahl der Pufferrahmen, die zur Verfügung stehen.

Wenn die Anzahl der Seiten um eins höher ist als die Anzahl der Pufferspeicher, dann wird bei den unter a) aufgezählten Strategien jedes Mal bei drohender Überfüllung des Puffers jene Stelle überschrieben, welche die Seite, die als nächstes benötigt wird, enthält. Effektiv wird dadurch die Pufferung alles verlangsamt, weil sie bei jeder Inanspruchnahme einen Speicherbereich löschen und neu beschreiben muss.

Bei der MRU-Strategie wird der Speicher „von hinten“ gelöscht, somit ist die Gefahr hier nicht gegeben.