

Mérési jegyzőkönyv – Adatbázisok Laboratórium

5. mérés: Relációs lekérdezések optimalizálása

Név:	Rittgasszer Ákos
Neptun kód:	Z8WK8D
Feladat kódja:	33-Video
Mérésvezető neve:	Tóth Kristóf
Mérés időpontja:	2019-12-04 14:15
Mérés helyszíne:	HSZK L
Megoldott feladatok:	1a, 1b, 2a, 2b, 2c, 2d, 2e, 3a, 3b, 3c, 3d
Elérhető pontszám (plusz pontok nélkül):	13p

Mérési feladatok megoldása

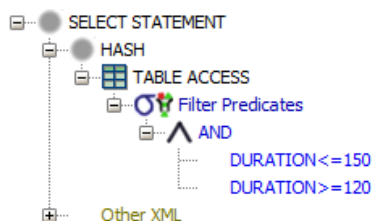
1. Feladat: 120 és 150 közötti hosszúságú könyvek

a. Részfeladat: lekérdezés végrehajtási tervének vizsgálata

A megoldáshoz használt SQL utasítás

```
SELECT DISTINCT title, fee  
FROM EDU_VIDEO.VIDEOS  
WHERE duration BETWEEN 120 AND 150;
```

Végrehajtási terv:



Magyarázat:

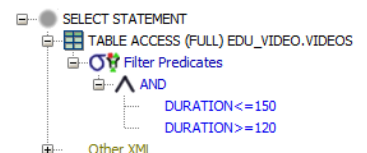
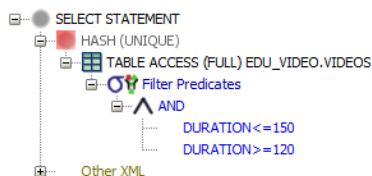
A lekérdezés kiválasztja a 120 és 150 közötti hosszúságú könyvek címét és kölcsönzési díját. Az egyediség biztosításához hash-t használ az adatbáziskezelő

b. Részfeladat: a módosított lekérdezés végrehajtási tervének összehasonlítása

A megoldáshoz használt SQL utasítás

```
SELECT DISTINCT title, fee, video_code  
FROM EDU_VIDEO.VIDEOS  
WHERE duration BETWEEN 120 AND 150;
```

Az a. és b. részfeladat végrehajtási tervének összehasonlítása



Magyarázat:

A lekérdezés ugyanazt csinálja mint az a részfeladatban, csak a video_code-ot is lekérdezi. Abbankülönbözik ez a lekérdezés az a részfeladatban látottól, hogy nem használ hash-t. A hasher azért nincsen szükség mert van egy indexe a táblának (VIDEO_CODE_UQ) amivel a video_codot lehet keresni, és a video_code egyedi attribútum.

2. Feladat: Indexek

a. Részfeladat: indexek

Az EDU_VIDEO_INDEX séma VIDEOS táblájának indexei:

- VIDEO_CODE_UQ(VIDEO_CODE)
- VIDEO_PK(VIDEO_ID)
- VIDEOS_DURATION_FEE_IDX(DURATION, FEE)

	INDEX_OWNER	INDEX_NAME	UNIQUENESS	STATUS	INDEX_TYPE	TEMPORARY	PARTITIONED	FUNCIDX_STATUS	JOIN_INDEX	COLUMNS
1	EDU_VIDEO_INDEX	VIDEO_PK	UNIQUE	VALID	NORMAL	N	NO	(null)	NO	VIDEO_ID
2	EDU_VIDEO_INDEX	VIDEO_CODE_UQ	UNIQUE	VALID	NORMAL	N	NO	(null)	NO	VIDEO_CODE
3	EDU_VIDEO_INDEX	VIDEOS_DURATION_FEE_IDX	NONUNIQUE	VALID	NORMAL	N	NO	(null)	NO	DURATION, FEE

b. Részfeladat: lekérdezés indexelt sémából

A megoldáshoz használt SQL utasítás

```
SELECT video_code, title, fee
FROM EDU_VIDEO_INDEX.VIDEOS
WHERE duration = 120 AND fee = 516;
```

Végrehajtási terv:

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				8
TABLE ACCESS	EDU_VIDEO_INDEX.VIDEOS	BY INDEX ROWID BATCHED	7	8
INDEX	EDU_VIDEO_INDEX.VIDEOS_DURATION_FEE_IDX	RANGE SCAN	7	1
Access Predicates				
AND				
DURATION=120				
FEE=516				
Other XML				

Keresés elsődleges index alapján

Becsült rekordszám: 7

Tényleges rekorszám: 98

c. Részfeladat: módosított lekérdezés

A megoldáshoz használt SQL utasítás

```
SELECT video_code, title, fee
FROM EDU_VIDEO_INDEX.VIDEOS
WHERE fee < 500 AND duration BETWEEN 120 AND 130;
```

Végrehajtási terv:

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				273
TABLE ACCESS	EDU_VIDEO_INDEX.VIDEOS	FULL	405	273
Filter Predicates				
AND				
FEE < 500				
DURATION <= 150				
DURATION >= 120				
Other XML				

Nem index segítségével történik a keresés

d. Részfeladat: módosított lekérdezés

A megoldáshoz használt SQL utasítás

```
SELECT video_code, title, fee  
FROM EDU_VIDEO_INDEX.VIDEOS  
WHERE fee < 500 AND duration BETWEEN 120 AND 150;
```

Végrehajtási terv:

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				150
TABLE ACCESS	EDU_VIDEO_INDEX.VIDEOS	BY INDEX ROWID BATCHED	144	150
INDEX	EDU_VIDEO_INDEX.VIDEOS_DURATION...	RANGE SCAN	144	7
Access Predicates				
AND				
DURATION >= 120				
DURATION <= 130				
FEE < 500				
Filter Predicates				
FEE < 500				
Other XML				

Megint index segítségével keres

e. Részfeladat: Összehasonlítás

Számosság c estben: 405

Számosság d estben: 144

A c lekérdezésben több rekordra teljesül a between feltétel mint a d ben. Összesen 1000 blokk van a táblában. Ha kisebb intervallumon belül teljesül a feltétel akkor kevesebb blokkot kell beolvasni a memóriába. Ha belefér az összes blokk a memóriába (d eset) akkor lehet indexek segítségével a keresni, ha nem akkor lineárisan kell.

23 BLOCKS

1000

3. Feladat:

a. Részfeladat:

A megoldáshoz használt SQL utasítás

```
SELECT member_code, name
FROM EDU_VIDEO.MEMBERS, EDU_VIDEO.LOANS
WHERE name LIKE 'M%' AND levelofmember = 'GOLD' AND
dateofbirth < DATE'1930-01-01' AND loans.member_id = members.member_id;
```

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				80 341
HASH JOIN				80 341
Access Predicates				
LOANS.MEMBER_ID=MEMBERS.MEMBER_ID				
TABLE ACCESS	EDU_VIDEO.MEMBERS	FULL	20	136
Filter Predicates				
AND				
DATEOFBIRTH<TO_DATE(' 1930-01-01 00:00:00', 'yyyy-mm-dd hh24:mi:ss')				
LEVELOFMEMBER='GOLD'				
NAME LIKE 'M%'				
TRUNC(INTERNAL_FUNCTION(DATEOFBIRTH))<TO_DATE(' 1930-01-01 00:00:00', 'yyyy-mm-dd hh24:mi:ss')				
TABLE ACCESS	EDU_VIDEO.LOANS	FULL	120000	205
Other XML				

Magyarázat:

Hash join-nal. A külsőbe a EDU_VIDEO.MEMBERS kerül 20 rekorddal, a belsőbe pedig a EDU_VIDEO.LOANS kerül 120000 rekorddal

b. Részfeladat: az indexelt sémába

A megoldáshoz használt SQL utasítás

```
SELECT member_code, name
FROM EDU_VIDEO_INDEX.MEMBERS, EDU_VIDEO_INDEX.LOANS
WHERE name LIKE 'M%' AND levelofmember = 'GOLD' AND
dateofbirth < DATE'1930-01-01' AND loans.member_id = members.member_id;
```

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				102 156
NESTED LOOPS				102 156
TABLE ACCESS	EDU_VIDEO_INDEX.MEMBERS	FULL	20	136
Filter Predicates				
AND				
DATEOFBIRTH<TO_DATE(' 1930-01-01 00:00:00', 'yyyy-mm-dd hh24:mi:ss')				
LEVELOFMEMBER='GOLD'				
NAME LIKE 'M%'				
TRUNC(INTERNAL_FUNCTION(DATEOFBIRTH))<TO_DATE(' 1930-01-01 00:00:00', 'yyyy-mm-dd hh24:mi:ss')				
INDEX	EDU_VIDEO_INDEX.LOANS_MEMBER_IDX	RANGE SCAN	5	1
Access Predicates				
LOANS.MEMBER_ID=MEMBERS.MEMBER_ID				
Other XML				

LOANS_MEMBER_IDX-et használ

c. Részfeladat: általánosítás

Ha indexelt sémát használunk join-hoz, akkor az illesztés feltételét tartalmazó indexet fogja használni egy nested loop algoritmussal

d. Részfeladat: kiegészítés

A megoldáshoz használt SQL utasítás

```
SELECT member_code, name, loans.loan_id, loans.video_id, loans.dateofcreation  
FROM EDU_VIDEO_INDEX.MEMBERS, EDU_VIDEO_INDEX.LOANS  
WHERE name LIKE 'M%' AND levelofmember = 'GOLD' AND  
dateofbirth < DATE'1930-01-01' AND loans.member_id = members.member_id;
```

Végrehajtási terv:

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				102 251
NESTED LOOPS				102 251
NESTED LOOPS				102 251
TABLE ACCESS	EDU_VIDEO_INDEX.MEMBERS	FULL		20 136
Filter Predicates				
AND				
DATEOFBIRTH < TO_DATE('1930-01-01 00:00:00', 'yyyy-mm-dd hh24:mi:ss')				
LEVELOFMEMBER = 'GOLD'				
NAME LIKE 'M%'				
TRUNC(INTERNAL_FUNCTION(DATEOFBIRTH)) < TO_DATE('1930-01-01 00:00:00', 'yyyy-mm-dd hh24:mi:ss')				
INDEX	EDU_VIDEO_INDEX.LOANS_MEMBER_IDX	RANGE SCAN		5 1
Access Predicates				
LOANS.MEMBER_ID = MEMBERS.MEMBER_ID				
TABLE ACCESS	EDU_VIDEO_INDEX.LOANS	BY INDEX ROWID		5 7
Other XML				

Kettő illesztési algoritmus van, egy nested loop join ami azonos a b részfeladatban levővel, és egy másik nested loop join ami az előző eredményéhez illeszt a LOANS táblát.