ORACLE

Index

Mag. Thomas Griesmayer

Index

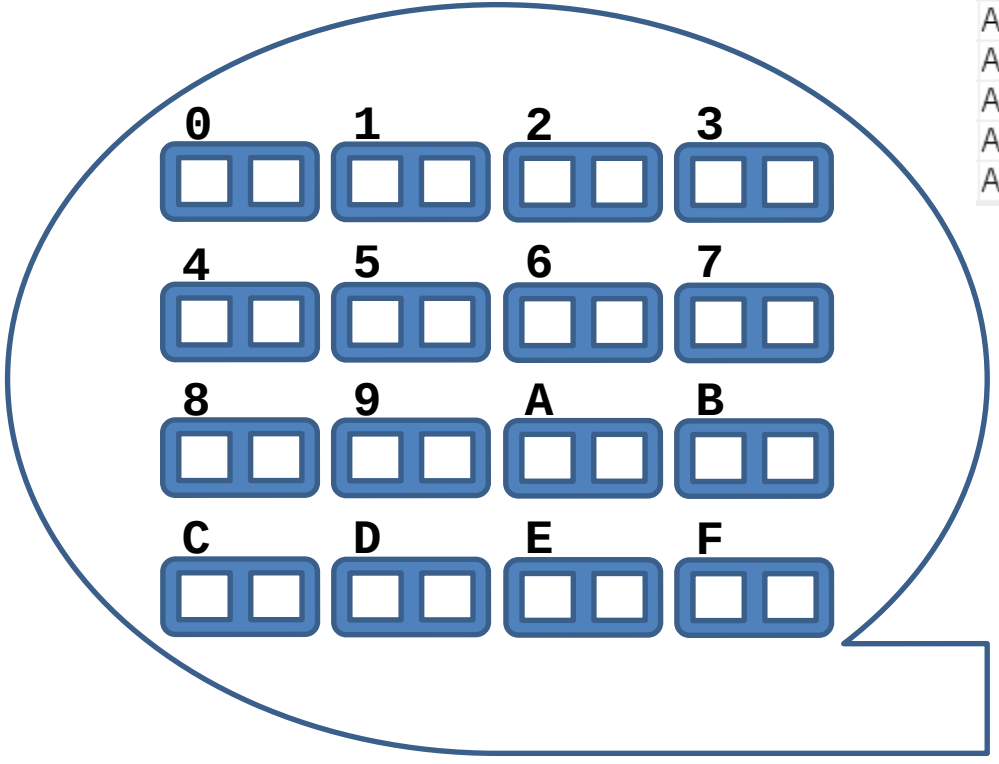
- Indexes are optional structures associated with tables and clusters that allow SQL statements to execute more quickly against a table.
- Just as an index in a manual helps you locate information faster than if there were no index, an Oracle Database index provides a faster access path to table data.
- You can use indexes without rewriting any queries.
- Your results are the same, but you see them more quickly.

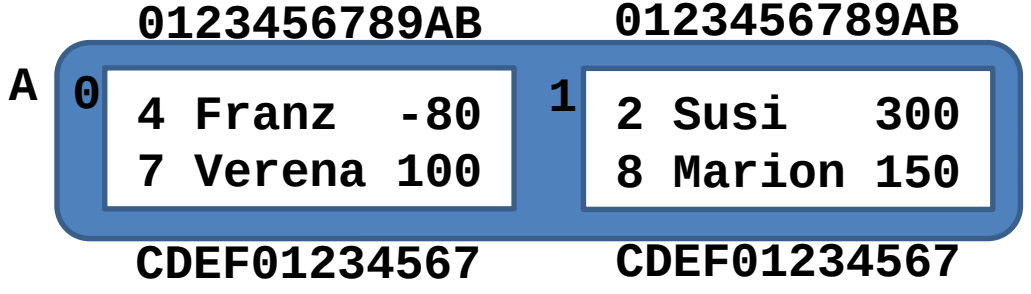
https://docs.oracle.com/cd/B19306_01/server.102/b14231/indexes.htm (15.9.2019)

ROWID

- The ROWID is the fastest way to access a single row.
- It consists of the absolute and relative adress.

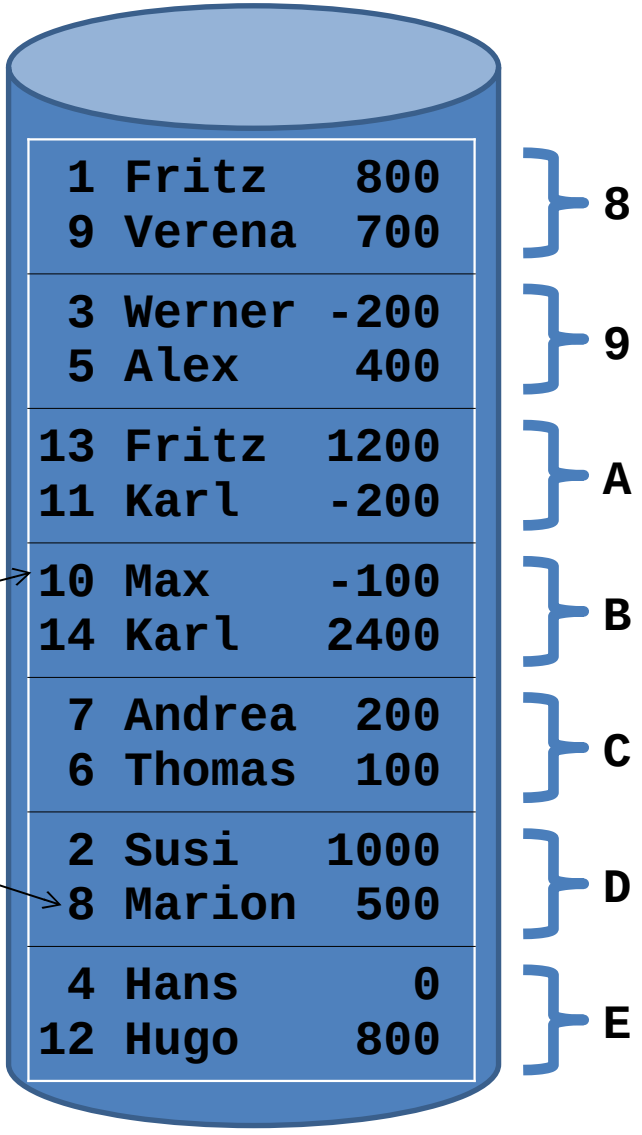
ROWID	CUSTOMER_ID	FIRST_NAME	BALANCE
AAAb3/AAEAAAFHTAAA	1	Fritz	200
AAAb3/AAEAAAFHTAAB	2	Susi	300
AAAb3/AAEAAAFHTAAC	3	Hans	150
AAAb3/AAEAAAFHTAAD	4	Franz	-80
AAAb3/AAEAAAFHTAAE	5	Maria	0
AAAb3/AAEAAAFHTAAF	6	Andrea	550
AAAb3/AAEAAAFHTAAG	7	Verena	100
AAAb3/AAEAAAFHTAAH	8	Marion	150





A00C => 7 Verena 100
A100 => 2 Susi 300

RID	C_ID	NAME	BALANCE
80	1	Fritz	€ 800
D0	2	Susi	€ 1000
90	3	Werner	€ -200
E0	4	Hans	€ 0
9C	5	Alex	€ 400
CC	6	Thomas	€ 100
C0	7	Andrea	€ 200
DC	8	Marion	€ 500
8C	9	Verena	€ 700
B0	10	Max	€ -100
AC	11	Karl	€ -200
EC	12	Hugo	€ 800
A0	13	Fritz	€ 1200
BC	14	Karl	€ 2400

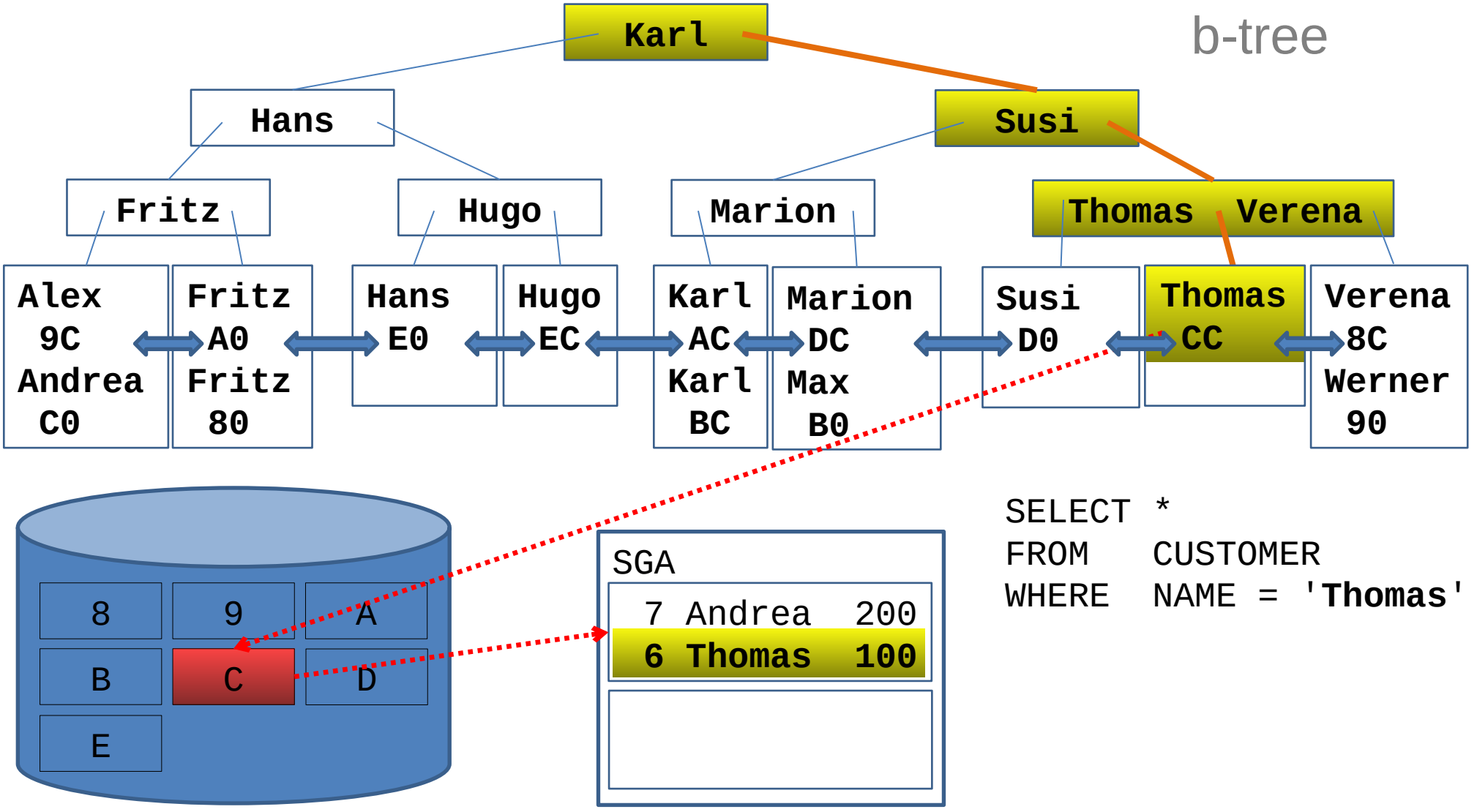


Physical address

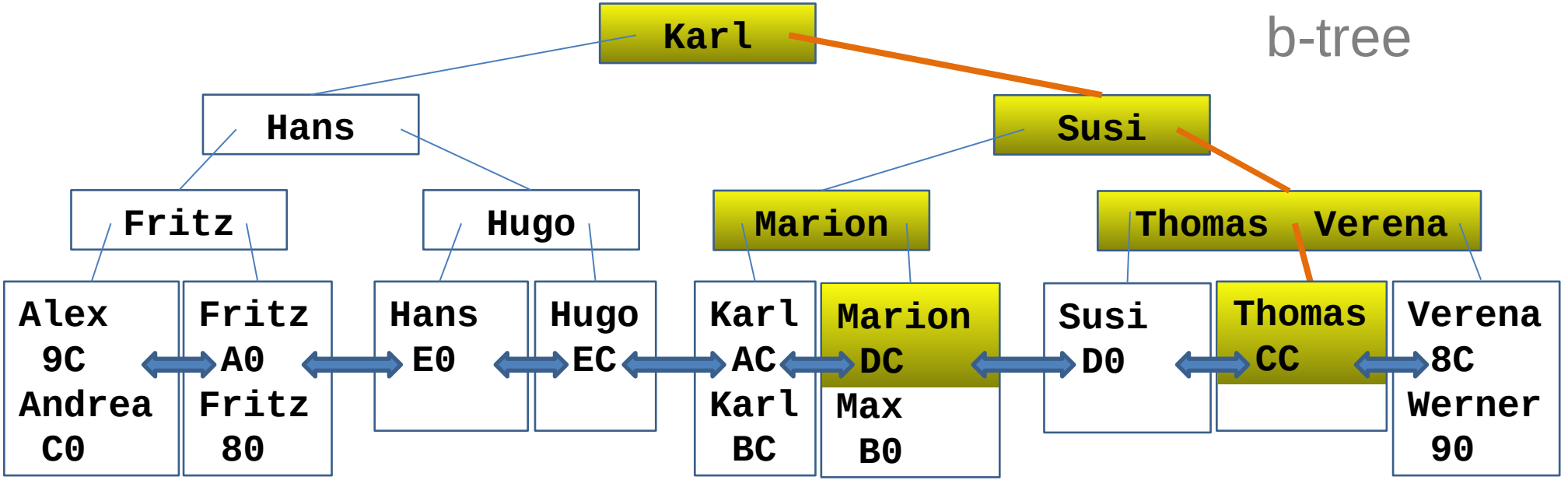
b-tree

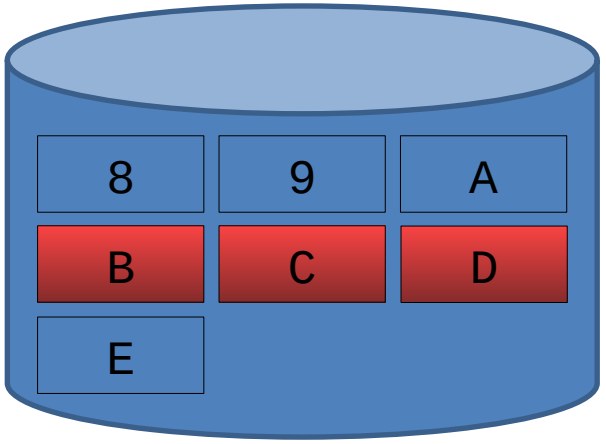
- By default, Oracle creates a b-tree index.
- Searching a specific line in a b-tree, Oracle walks the branches until it gets to the node with the relevant data.

```
CREATE INDEX IDX_NAME ON CUSTOMER (NAME);
```


b-tree
Karl
Hans
Susi
Fritz
Hugo
Marion
Thomas
Verena
Alex
9C
Andrea
C0
Fritz
A0
Fritz
80
Hans
E0
Hugo
EC
Karl
AC
Karl
BC
Marion
DC
Max
B0
Susi
D0
Thomas
CC
Verena
8C
Werner
90
8
9
A
B
C
D
E
SGA
7 Andrea 200
6 Thomas 100
SELECT *
FROM CUSTOMER
WHERE NAME = 'Thomas'

b-tree





SGA

2	**Susi**	**1000**
8	**Marion**	**500**
10	**Max**	**-100**
14	Karl	2400
7	Andrea	200
6	**Thomas**	**100**

```
SELECT *
FROM   CUSTOMER
WHERE  NAME between
          'Marion' and
          'Thomas'
```

Visualization

Bitmap

- Bitmap indexes have traditionally been considered to work well for low-cardinality columns, which have a modest number of distinct values, either absolutely, or relative to the number of records that contain the data.
- The extreme case of low cardinality is Boolean data (e.g., does a resident in a city have internet access?), which has two values, True and False.
- Bitmap indexes use bit arrays (commonly called bitmaps) and answer queries by performing bitwise logical operations on these bitmaps.

```
CREATE BITMAP INDEX IDX_GENDER ON CUSTOMER (GENDER);
```

https://en.wikipedia.org/wiki/Bitmap_index (15.9.2019)

Bitmap

RID	C_ID	NAME	BALANCE	GENDER
80	1	Fritz	€ 800	m
D0	2	Susi	€ 1000	f
90	3	Werner	€ -200	m
E0	4	Hans	€ 0	m
9C	5	Alex	€ 400	m
CC	6	Thomas	€ 100	m
C0	7	Andrea	€ 200	f
DC	8	Marion	€ 500	f
8C	9	Verena	€ 700	f
B0	10	Max	€ -100	m
AC	11	Karl	€ -200	m
EC	12	Hugo		
A0	13	Fritz		
BC	14	Karl		

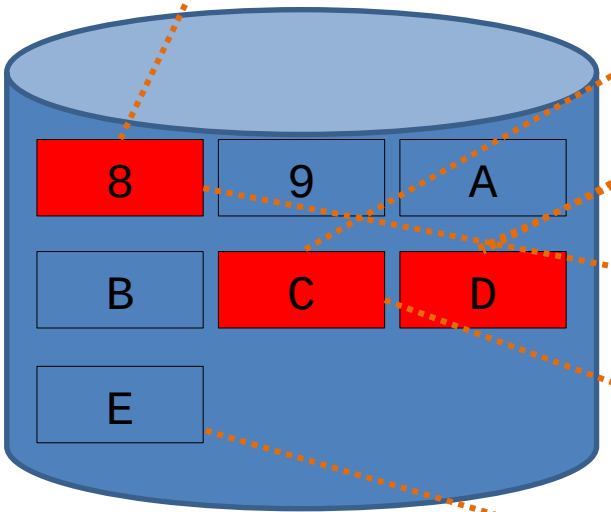
	80	8C	90	9C	A0	AC	B0	BC	C0	CC	D0	DC	E0	EC
m	1	0	1	1	1	1	1	1	0	1	0	0	1	1
f	0	1	0	0	0	0	0	0	1	0	1	1	0	0

Bitmap

```
SELECT count(*)
FROM   CUSTOMER
WHERE  GENDER = 'm';
```

```
SELECT *
FROM   customer
WHERE  gender = 'f';
```

	80	8C	90	9C	A0	AC	B0	BC	C0	CC	D0	DC	E0	EC
m	1	0	1	1	1	1	1	1	0	1	0	0	1	1
f	0	1	0	0	0	0	0	0	1	0	1	1	0	0



SGA

1	Fritz	800	m
9	**Verena**	**700**	**f**
7	**Andrea**	**200**	**f**
6	Thomas	100	m
2	**Susi**	**1000**	**f**
8	**Marion**	**500**	**f**

Hash

- Hashing is the transformation of an database datatype (**VARCHAR, DECIMAL**) into a usually shorter key that represents the original data.
- It is faster to find the data by using the hashed key to identify the **ROWID** and the relevant data.

- **There is no hash index in oracle.**

RID	C_ID	NAME	BALANCE	GENDER	1st char	NR	NR%10
80	1	Fritz	€ 800	m	F	6	6
D0	2	Susi	€ 1000	f	S	19	9
90	3	Werner	€ -200	m	W	23	3
E0	4	Hans	€ 0	m	H	8	8
9C	5	Alex	€ 400	m	A	1	1
CC	6	Thomas	€ 100	m	T	20	0
C0	7	Andrea	€ 200	f	A	1	1
DC	8	Marion	€ 500	f	M	13	3
8C	9	Verena	€ 700	f	V	22	2
B0	10	Max	€ -100	m	M	13	3
AC	11	Karl	€ -200	m	K	11	1
EC	12	Hugo	€ 800	m	H	8	8
A0	13	Fritz	€ 1200	m	F	6	6
BC	14	Karl	€ 2400	m	K	11	1

Hash

INDEX

HASH	RID	RID	RID	RID
0	CC			
1	9C	C0	AC	BC
2	8C			
3	90	DC	B0	
4				
5				
6	**80**	**A0**		
7				
8	E0	EC		
9	D0			

```
SELECT *
FROM   customer
WHERE  name = 'Fritz'
```

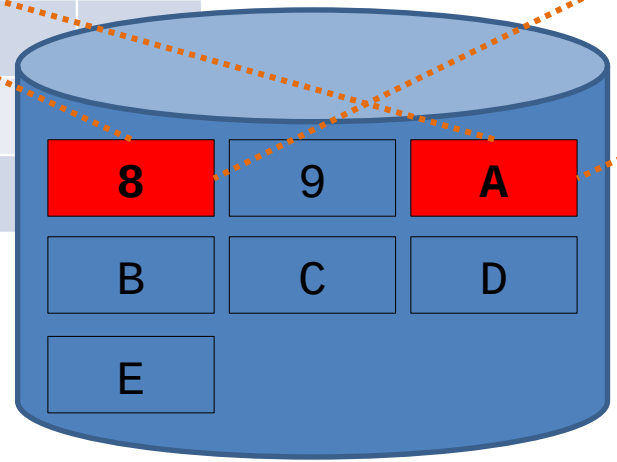
CALCULATE HASH

1st char	nr	nr%10
F	6	**6**

SGA

```
 1 Fritz   800
 9 Verena  700

13 Fritz  1200
11 Karl   -200
```



CUSTOMER

C_ID	NAME	BALANCE	GENDER
1	Fritz	€ 800	m
13	Fritz	€ 1200	m