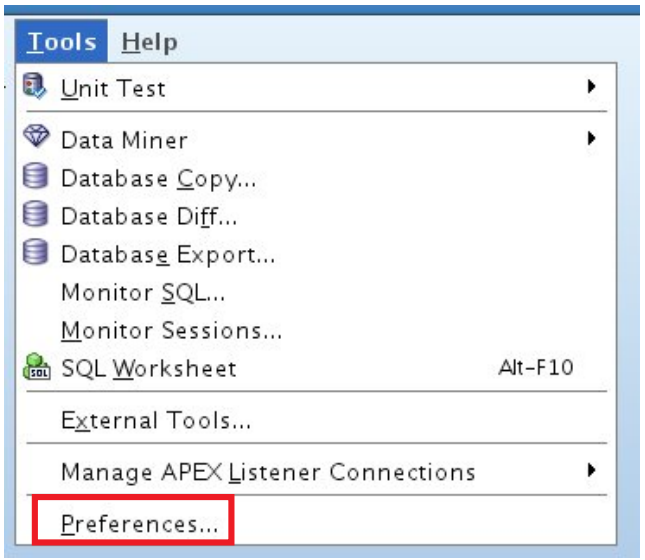
ORACLE

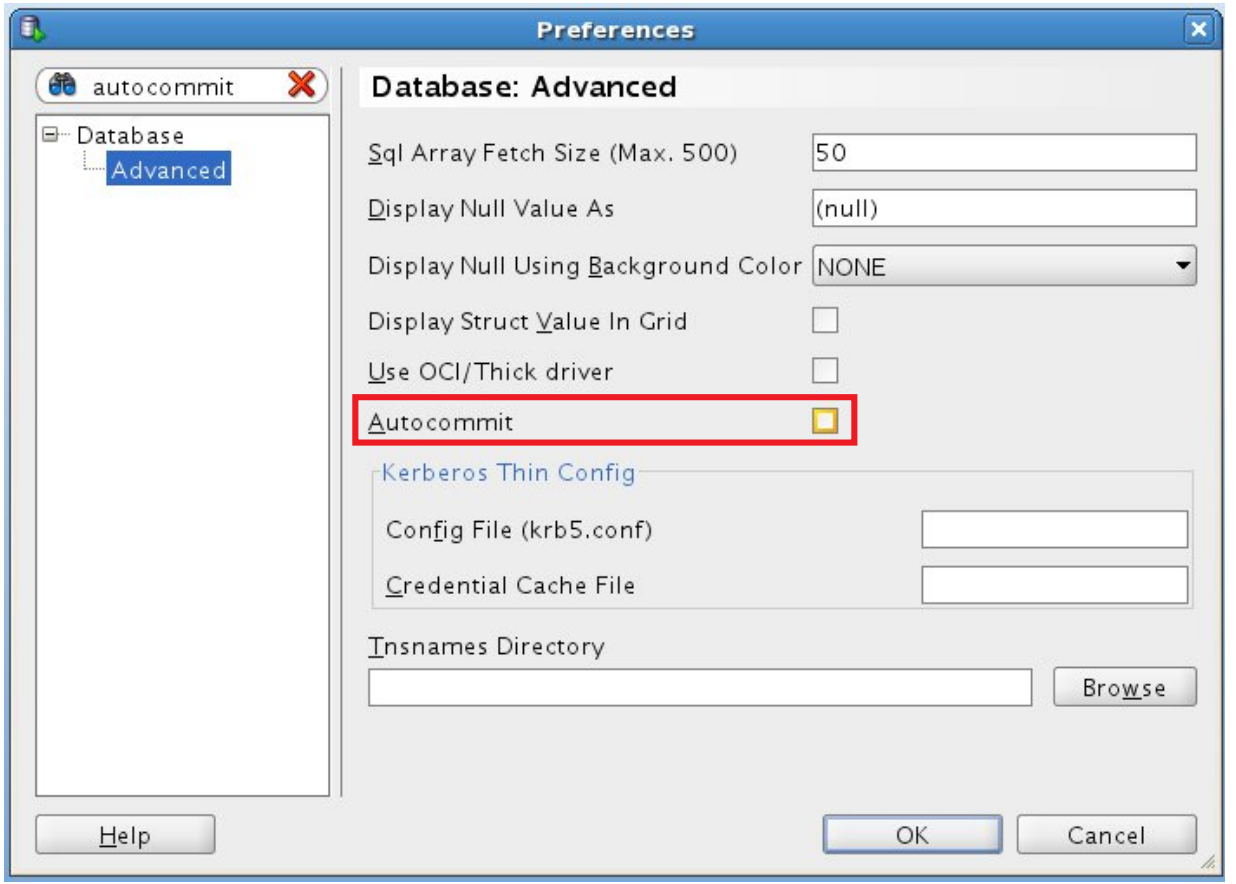
Redo Undo

Mag. Thomas Griesmayer

Autocommit

- The autocommit option COMMITS every DML statement after its execution.





Transaction

- The use of transactions is one of the most important ways that a database management system differs from a file system.
- A transaction is a logical, atomic unit of work that contains one or more SQL statements.
- A transaction groups SQL statements so that they are either all committed, which means they are applied to the database, or all rolled back, which means they are undone from the database.
- Oracle Database assigns every transaction a unique identifier called a transaction ID.

https://docs.oracle.com/database/121/CNCPT/transact.htm#CNCPT117 (7.9.2019)

ACID

- All Oracle transactions obey the basic properties of a database transaction, known as ACID properties.
- ACID is an acronym for the following:
  - **A**tomicity - All tasks of a transaction are performed or none of them are.
  - **C**onsistency - The transaction takes the database from one consistent state to another consistent state.
  - **I**solation - The effect of a transaction is not visible to other transactions until the transaction is committed.
  - **D**urability - Changes made by committed transactions are permanent. After a transaction completes, the database ensures through its recovery mechanisms that changes from the transaction are not lost.

https://docs.oracle.com/database/121/CNCPT/transact.htm#CNCPT117 (7.9.2019)

Transaction ID

```
SELECT *
FROM   CUSTOMER;

UPDATE CUSTOMER
SET    BALANCE = 550
WHERE  CUSTOMER_ID = 6;

UPDATE CUSTOMER
SET    BALANCE = 150
WHERE  CUSTOMER_ID = 8;

COMMIT;

INSERT INTO CUSTOMER
VALUES (9, 'Martina', 800);

DELETE FROM CUSTOMER
WHERE  CUSTOMER_ID = 1;

ROLLBACK;
```

```
SELECT XID
FROM   V$TRANSACTION;
no rows selected
```

```
SELECT XID
FROM   V$TRANSACTION;
0400200015250000
```

```
SELECT XID
FROM   V$TRANSACTION;
09000000B0250000
```

ROWID

- For each row in the database, the ROWID pseudocolumn returns the address of the row.
- Oracle Database rowid values contain information necessary to locate a row:
  - The data object number of the object
  - The data block in the datafile in which the row resides
  - The position of the row in the data block (first row is 0)
  - The datafile in which the row resides (first file is 1). The file number is relative to the tablespace.

http://docs.oracle.com/cd/B19306_01/server.102/b14200/pseudocolumns008.htm (7.9.2019)

ROWID

- Rowid values have several important uses:
  - They are the fastest way to access a single row.
  - They can show you how the rows in a table are stored.
  - They are unique identifiers for rows in a table.
  - You should not use ROWID as the primary key of a table.
  - If you delete and reinsert a row with the Import and Export utilities, for example, then its rowid may change.
  - If you delete a row, then Oracle may reassign its rowid to a new row inserted later.

http://docs.oracle.com/cd/B19306_01/server.102/b14200/pseudocolumns008.htm (7.9.2019)

ROWID

```
SELECT ROWID,
       cus.*
FROM   CUSTOMER cus;
```

Script Output | Query R...

SQL | All Rows Fetched: 8 in 0.018 seconds

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

```
SELECT ROWID,
       cus.*
FROM   CUSTOMER cus
WHERE  ROWID = 'AAAb3/AAEAAAFHTAAB';
```

Script Output | Query R...

SQL | All Rows Fetched: 1 in 0.001 seconds

	ROWID	CUSTOMER_ID	FIRST_NAME	BALANCE
1	AAAb3/AAEAAAFHTAAB	2	Susi	300

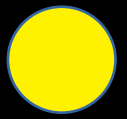
```
SELECT *
FROM   CUSTOMER;
```

TRANSFER

RID	FROM	TO	DATE	AMOUNT

CUSTOMER

RID	CID	NAME	BALANCE
A0	1	Fritz	300
C1	2	Marion	400
A1	3	Andrea	200
B0	4	Hans	-300
B1	5	Verena	1000
C0	6	Thomas	800



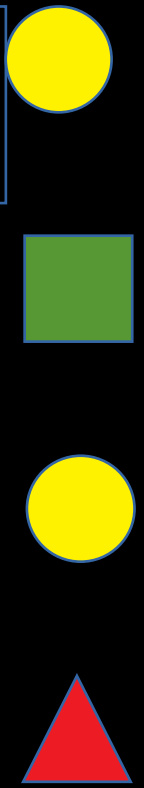
REDO

TID	RID	VALUE

UNDO

TID	RID	VALUE

```
UPDATE  CUSTOMER
SET     BALANCE = BALANCE-50
WHERE   CUSTOMER_ID = 3;
```



TRANSFER

RID	FROM	TO	DATE	AMOUNT

CUSTOMER

RID	CID	NAME	BALANCE
A0	1	Fritz	300
C1	2	Marion	400
A1	3	Andrea	200
B0	4	Hans	-300
B1	5	Verena	1000
C0	6	Thomas	800

REDO

TID	RID	VALUE

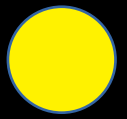
UNDO

TID	RID	VALUE

```
UPDATE  CUSTOMER
SET     BALANCE = BALANCE-200
WHERE   CUSTOMER_ID = 4;
```

TRANSFER

RID	FROM	TO	DATE	AMOUNT



CUSTOMER

RID	CID	NAME	BALANCE
A0	1	Fritz	300
C1	2	Marion	400
A1	3	Andrea	200
B0	4	Hans	-300
B1	5	Verena	1000
C0	6	Thomas	800

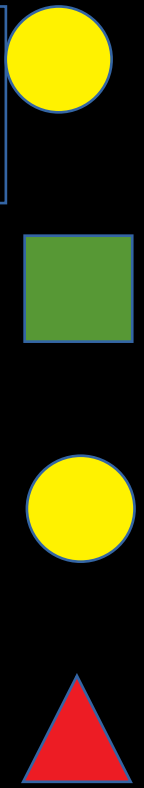
REDO

TID	RID	VALUE

UNDO

TID	RID	VALUE

```
UPDATE  CUSTOMER
SET     BALANCE = BALANCE+50
WHERE   CUSTOMER_ID = 1;
```



TRANSFER

RID	FROM	TO	DATE	AMOUNT

CUSTOMER

RID	CID	NAME	BALANCE
A0	1	Fritz	300
C1	2	Marion	400
A1	3	Andrea	200
B0	4	Hans	-300
B1	5	Verena	1000
C0	6	Thomas	800

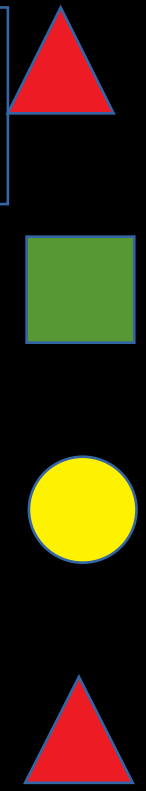
REDO

TID	RID	VALUE

UNDO

TID	RID	VALUE

```
UPDATE  CUSTOMER
SET     BALANCE = BALANCE-5
WHERE   CUSTOMER_ID = 2;
```



TRANSFER

RID	FROM	TO	DATE	AMOUNT

CUSTOMER

RID	CID	NAME	BALANCE
A0	1	Fritz	300
C1	2	Marion	400
A1	3	Andrea	200
B0	4	Hans	-300
B1	5	Verena	1000
C0	6	Thomas	800

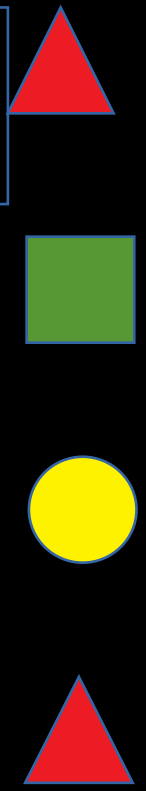
REDO

TID	RID	VALUE

UNDO

TID	RID	VALUE

```
UPDATE  CUSTOMER
SET     BALANCE = BALANCE+5
WHERE   CUSTOMER_ID = 6;
```



TRANSFER

RID	FROM	TO	DATE	AMOUNT

CUSTOMER

RID	CID	NAME	BALANCE
A0	1	Fritz	300
C1	2	Marion	400
A1	3	Andrea	200
B0	4	Hans	-300
B1	5	Verena	1000
C0	6	Thomas	800

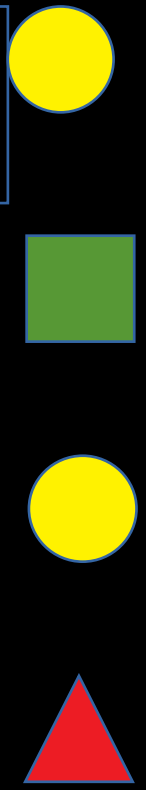
REDO

TID	RID	VALUE

UNDO

TID	RID	VALUE

```
INSERT INTO TRANSFER
VALUES (3, 1, SYSDATE, 50);
```



TRANSFER

RID	FROM	TO	DATE	AMOUNT

CUSTOMER

RID	CID	NAME	BALANCE
A0	1	Fritz	300
C1	2	Marion	400
A1	3	Andrea	200
B0	4	Hans	-300
B1	5	Verena	1000
C0	6	Thomas	800

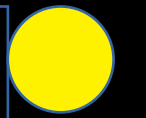
REDO

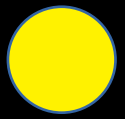
TID	RID	VALUE

UNDO

TID	RID	VALUE

```
COMMIT;
```





TRANSFER

RID	FROM	TO	DATE	AMOUNT

CUSTOMER

RID	CID	NAME	BALANCE
A0	1	Fritz	300
C1	2	Marion	400
A1	3	Andrea	200
B0	4	Hans	-300
B1	5	Verena	1000
C0	6	Thomas	800

REDO

TID	RID	VALUE

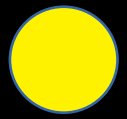
UNDO

TID	RID	VALUE

```
UPDATE  CUSTOMER
SET     BALANCE = BALANCE+200
WHERE   CUSTOMER_ID = 5;
```

TRANSFER

RID	FROM	TO	DATE	AMOUNT



CUSTOMER

RID	CID	NAME	BALANCE
A0	1	Fritz	300
C1	2	Marion	400
A1	3	Andrea	200
B0	4	Hans	-300
B1	5	Verena	1000
C0	6	Thomas	800

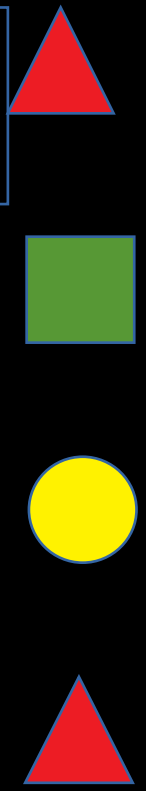
REDO

TID	RID	VALUE

UNDO

TID	RID	VALUE

```
INSERT INTO TRANSFER
VALUES (2, 6, SYSDATE, 5);
```



TRANSFER

RID	FROM	TO	DATE	AMOUNT

CUSTOMER

RID	CID	NAME	BALANCE
A0	1	Fritz	300
C1	2	Marion	400
A1	3	Andrea	200
B0	4	Hans	-300
B1	5	Verena	1000
C0	6	Thomas	800

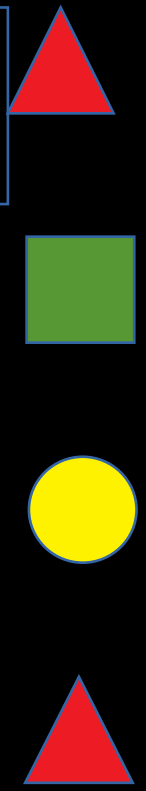
REDO

TID	RID	VALUE

UNDO

TID	RID	VALUE

```
ROLLBACK;
```



TRANSFER

RID	FROM	TO	DATE	AMOUNT

CUSTOMER

RID	CID	NAME	BALANCE
A0	1	Fritz	300
C1	2	Marion	400
A1	3	Andrea	200
B0	4	Hans	-300
B1	5	Verena	1000
C0	6	Thomas	800

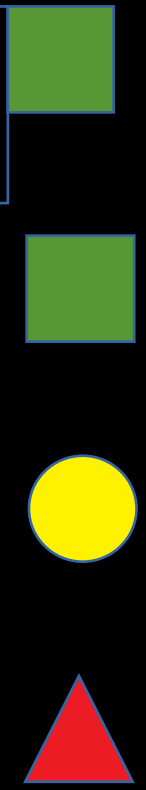
REDO

TID	RID	VALUE

UNDO

TID	RID	VALUE

```
INSERT INTO TRANSFER
VALUES (4, 5, SYSDATE, 200);
```



TRANSFER

RID	FROM	TO	DATE	AMOUNT

CUSTOMER

RID	CID	NAME	BALANCE
A0	1	Fritz	300
C1	2	Marion	400
A1	3	Andrea	200
B0	4	Hans	-300
B1	5	Verena	1000
C0	6	Thomas	800

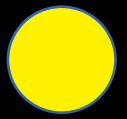
REDO

TID	RID	VALUE

UNDO

TID	RID	VALUE

```
COMMIT;
```



TRANSFER

RID	FROM	TO	DATE	AMOUNT

CUSTOMER

RID	CID	NAME	BALANCE
A0	1	Fritz	300
C1	2	Marion	400
A1	3	Andrea	200
B0	4	Hans	-300
B1	5	Verena	1000
C0	6	Thomas	800

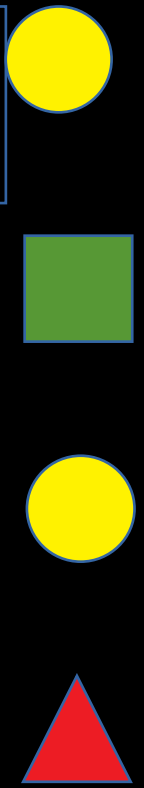
REDO

TID	RID	VALUE

UNDO

TID	RID	VALUE

```
UPDATE  CUSTOMER
SET     BALANCE = BALANCE-5
WHERE   CUSTOMER_ID = 2;
```



TRANSFER

RID	FROM	TO	DATE	AMOUNT

CUSTOMER

RID	CID	NAME	BALANCE
A0	1	Fritz	300
C1	2	Marion	400
A1	3	Andrea	200
B0	4	Hans	-300
B1	5	Verena	1000
C0	6	Thomas	800

REDO

TID	RID	VALUE

UNDO

TID	RID	VALUE