

DATABASE SYSTEMS

COURSE

2021/2022

SESSION 11

SESSION 11

- INDEXES
- TRIGGER

Query on table with no index

MySQL/Oracle

```
SELECT * FROM moviebig WHERE movie_id = 453;
```

	MOVIE_ID	TITLE
1	114	Pretty Woman
2	115	The Big Lebowski
3	116	Match Point
176	453	A Beautiful Mind
4801	117251	White House Down
4802	117263	Olympus Has Fallen
4803	117942	Girls Gone Dead

CREATE INDEX Syntax

MySQL/Oracle

```
CREATE INDEX IndexName ON table(Field1,Field2,...);
```

MySQL/Oracle

```
CREATE UNIQUE INDEX IndexName ON table(Field1,Field2,...);
```

NOTE: Unique Index are automatically created at Primary Key creation

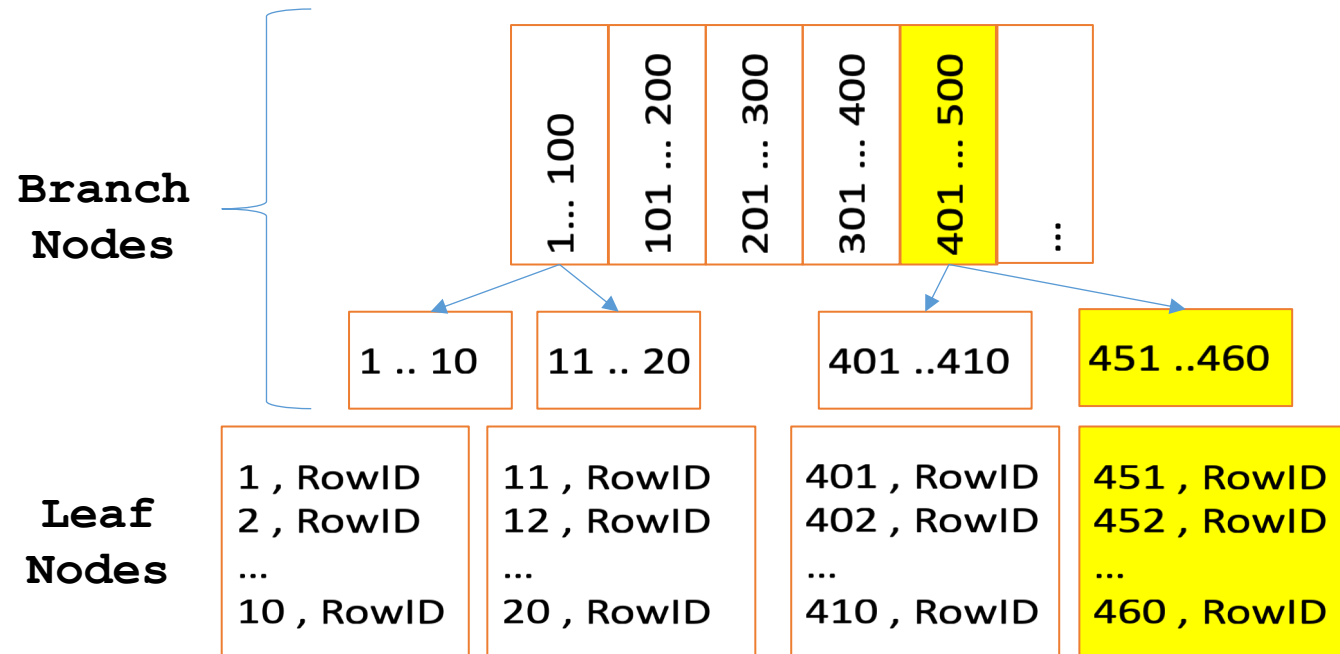
Query on a indexed column

MySQL/Oracle

```
CREATE INDEX idx_movieid  
ON moviebig(movie_id);
```

MySQL/Oracle

```
SELECT * FROM moviebig  
WHERE movie_id = 453;
```



INDEXES

CREATE INDEX Example

MySQL/Oracle

```
SELECT * FROM moviebig  
WHERE movie_id = 453;
```

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	102
TABLE ACCESS	MOVIEBIG	FULL	1	102
Filter Predicates				
MOVIE_ID=453				
Other XML				

MySQL/Oracle

```
CREATE INDEX idx_movieid  
ON moviebig(movie_id) ;
```

MySQL/Oracle

```
SELECT * FROM moviebig  
WHERE movie_id = 453;
```

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	2
TABLE ACCESS	MOVIEBIG	BY INDEX ROWI...	1	2
INDEX	IDX_MOVIEID	RANGE SCAN	1	1
Access Predicates				
MOVIE_ID=453				

DROP INDEX Syntax

MySQL

```
ALTER TABLE table DROP INDEX IndexName
```

```
DROP INDEX IndexName ON table
```

Oracle

```
DROP INDEX IndexName
```

DROP INDEX Example

MySQL

```
ALTER TABLE moviebig DROP INDEX idx_movieid;  
DROP INDEX idx_movieid ON movie;
```

Oracle

```
DROP INDEX idx_movieid;
```


SESSION 11

- INDEXES
- **TRIGGER**

CREATE TRIGGER Syntax

MySQL/Oracle

```
CREATE TRIGGER TriggerName
EXECUTIONWHEN TRIGGEREVENT ON table
FOR EACH ROW
DECLARE
    -- Variable Declaration
BEGIN
    -- Code to executed
END;
```

EXECUTIONWHEN takes on of the following Values:

- **BEFORE**
- **AFTER**

TRIGGEREVENT takes on of the following values:

- **DELETE**
- **INSERT**
- **UPDATE**
- *Oracle allows more than one separated by "OR". Mysql does not.*

CREATE TRIGGER Example

MySQL

```
CREATE TABLE paudit (  
  modifiername VARCHAR(50),  
  modificationdate DATE,  
  previousname VARCHAR(50),  
  newname VARCHAR(50));
```

Oracle

```
CREATE TABLE paudit (  
  modifiername VARCHAR2(50),  
  modificationdate DATE,  
  previousname VARCHAR2(50),  
  newname VARCHAR2(50));
```

CREATE TRIGGER Example

MySQL

```
CREATE TRIGGER auditpeople BEFORE UPDATE ON people
FOR EACH ROW
BEGIN
    INSERT INTO paudit VALUES (USER(), SYSDATE(), OLD.Name, NEW.Name) ;
END ;
```

Oracle

```
CREATE TRIGGER auditpeople BEFORE UPDATE ON people
FOR EACH ROW
BEGIN
    INSERT INTO paudit VALUES (USER, SYSDATE, :OLD.Name, :NEW.Name) ;
END ;
```

CREATE TRIGGER Example

MySQL/Oracle

```
SELECT * FROM people;
```

PERSONID	NAME	LASTNAME	DOCID
1	Peter	Castle	6865433
2	Vicky	April	7353156

MySQL/Oracle

```
SELECT * FROM paudit;
```

MODIFIERNAME	MODIFICATI...	PREVIOUS...	NEWNAME
--------------	---------------	-------------	---------

MySQL/Oracle

```
UPDATE people SET Name='Pedro' WHERE personid=1;
```

PERSONID	NAME	LASTNAME	DOCID
1	Pedro	Castle	6865433
2	Vicky	April	7353156

MODIFIER...	MODIFICATIONDATE	PREVIOUSNAME	NEWNAME
PAVEL	05/08/21	Peter	Pedro

DROP TRIGGER Syntax

MySQL/Oracle

```
DROP TRIGGER TriggerName
```

DROP TRIGGER Example

MySQL/Oracle

```
DROP TRIGGER auditpeople;
```