



Constraint Optimization or *The Difference One Comma Can Make*

Oren Nakdimon

www.db-oriented.com

✉ oren@db-oriented.com

☎ +972-54-4393763

🐦 @DBoriented

**Can you
read this
query easily?
If not, please
sit closer**

```
SELECT DEPARTMENT_ID,  
       COUNT(*) NUMBER_OF_EMPLOYEES,  
       AVG(SALARY) AVG_SALARY  
FROM EMPLOYEES  
GROUP BY DEPARTMENT_ID  
HAVING MAX(SALARY) > 2 * MIN(SALARY)
```



dbORIENTED



Follow @DBoriented

WHO AM I?

THINGS TO DO TODAY

Date 1993 COMPLETED

- 1) _____
- 2) Start developing
- 3) _____
- 4) in ORACLE6 +
- 5) _____
- 6) SQL*Forms 3.0
- 7) _____
- 8) +Oracle*CASE
- 9) _____
- 10) 5.0

<http://db-oriented.com>

500+ Technical Experts Helping Peers Globally

ORACLE[®]
ACE Program



ORACLE[®]
ACE Director



ORACLE[®]
ACE



ORACLE[®]
ACE Associate

3 Membership Tiers

- Oracle ACE Director
- Oracle ACE
- Oracle ACE Associate

bit.ly/OracleACEProgram

Connect:

✉ oracle-ace_ww@oracle.com

f Facebook.com/oracleaces

t @oracleace



Nominate yourself or someone you know: acenomination.oracle.com

Oracle Cloud Infrastructure

New Free Tier

oracle.com/gbtour

Always Free

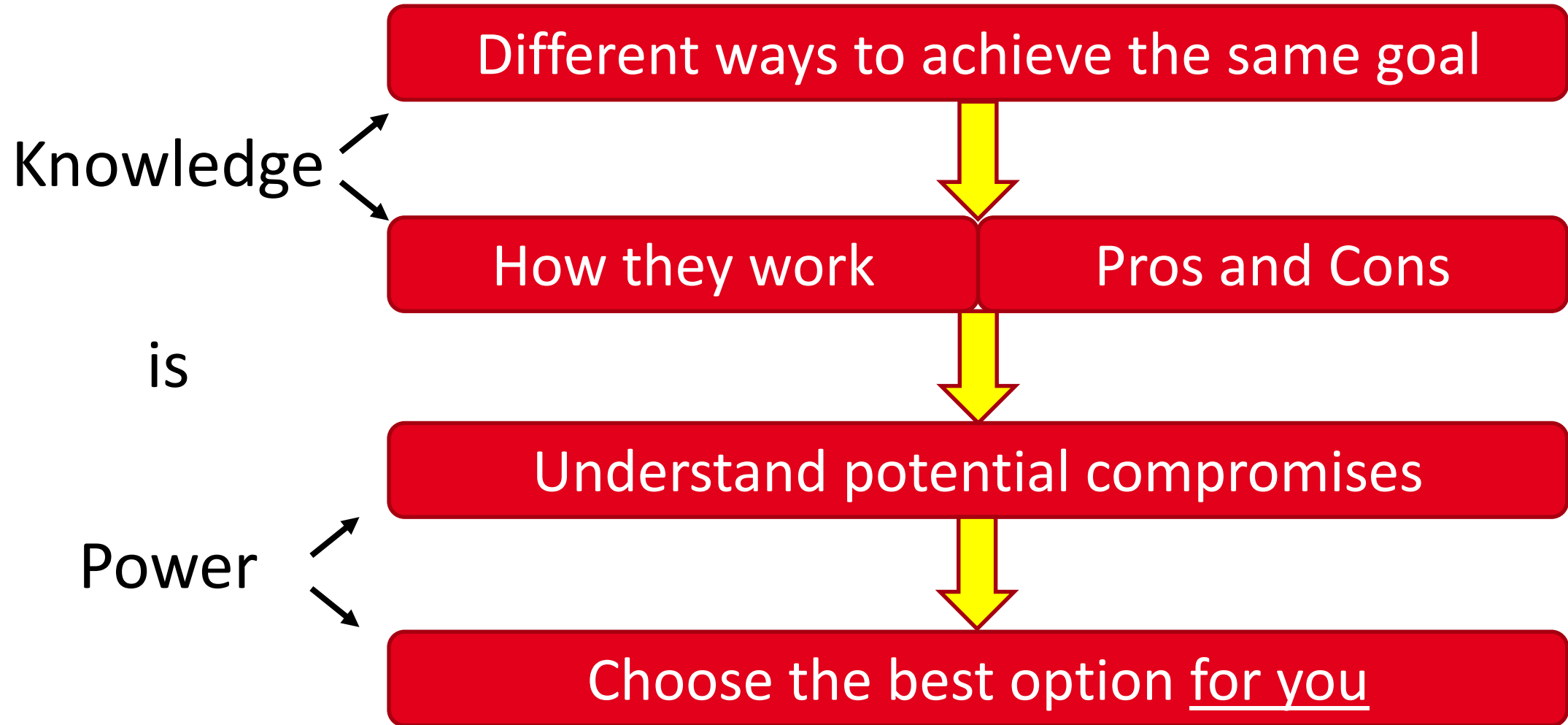
Services you can use for unlimited time

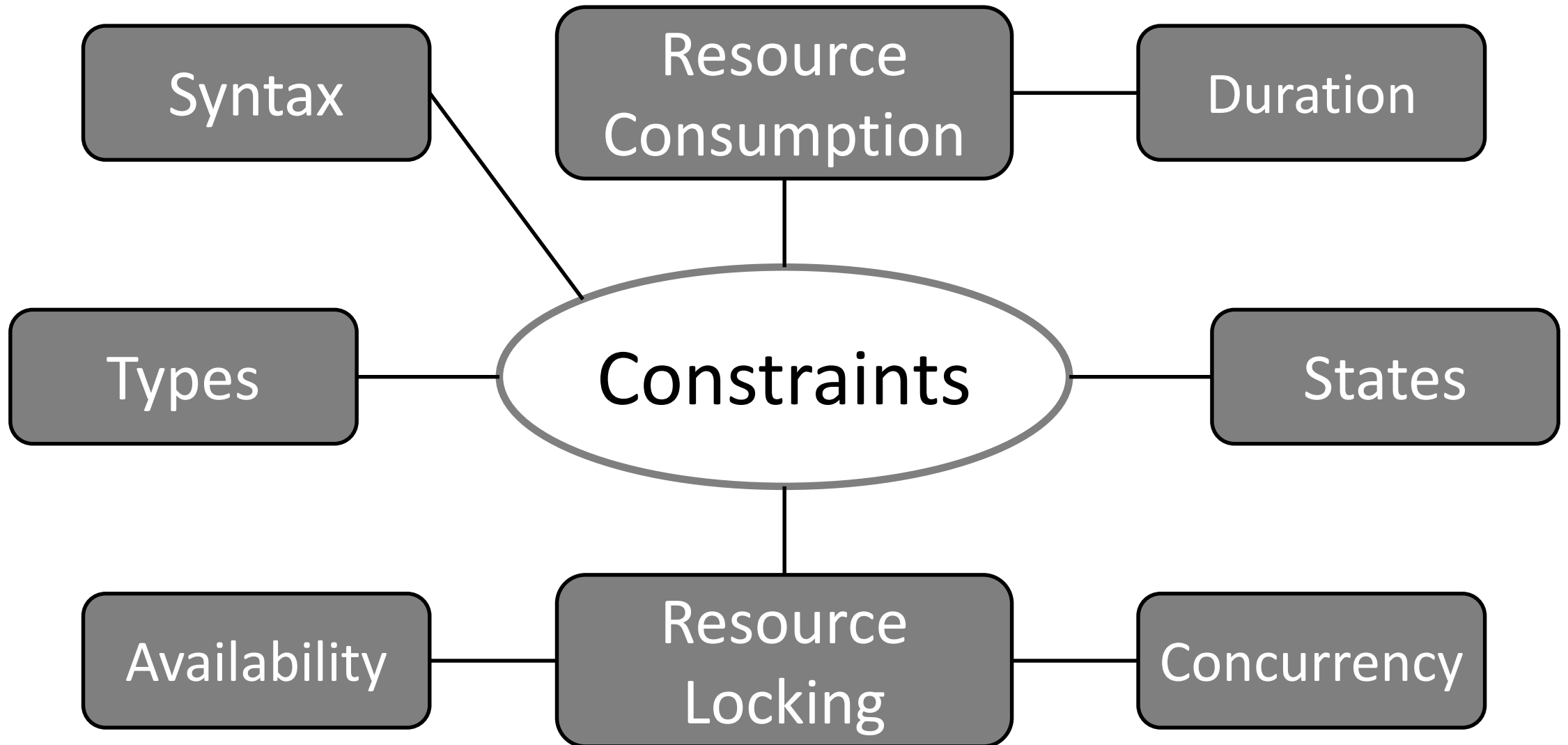


30-Day Free Trial

Free credits you can use for more services







A decorative vertical bar on the left side of the slide, featuring a gradient from light pink to dark red. It contains several red circles of varying sizes, with the largest one at the top left and smaller ones scattered below and to the right.

CONSTRAINT CREATION SYNTAX

Constraint Creation Syntax

**Inline
Constraints**

**Out-of-Line
Constraints**

Constraint Creation Syntax

**Inline
(column-level)
Constraints**

**Out-of-Line
(table-level)
Constraints**

No difference once the
constraint creation is over

Constraint Creation – Examples

```
CREATE TABLE EMPLOYEES (  
  EMPLOYEE_ID      NUMBER(6),  
  FIRST_NAME       VARCHAR2(20),  
  LAST_NAME        VARCHAR2(25) CONSTRAINT EMP_LAST_NAME_NN NOT NULL,  
  PHONE_NUMBER     VARCHAR2(20),  
  HIRE_DATE        DATE NOT NULL,  
  JOB_ID           VARCHAR2(10) NOT NULL,  
  SALARY           NUMBER(8,2),  
  COMMISSION_PCT  NUMBER(2,2),  
  MANAGER_ID       NUMBER(6),  
  DEPARTMENT_ID   NUMBER(4) CONSTRAINT EMP_DEPT_FK REFERENCES DEPARTMENTS,  
  CONSTRAINT EMP_JOB_FK FOREIGN KEY (JOB_ID) REFERENCES JOBS (JOB_ID),  
  CHECK (SALARY > 0)  
);
```

Inline Constraints

```
CREATE TABLE EMPLOYEES (  
  EMPLOYEE_ID    NUMBER(6),  
  FIRST_NAME     VARCHAR2(20),  
  LAST_NAME      VARCHAR2(25) CONSTRAINT EMP_LAST_NAME_NN NOT NULL,  
  PHONE_NUMBER   VARCHAR2(20),  
  HIRE_DATE      DATE NOT NULL,  
  JOB_ID         VARCHAR2(10) NOT NULL,  
  SALARY         NUMBER(8,2),  
  COMMISSION_PCT NUMBER(2,2),  
  MANAGER_ID     NUMBER(6),  
  DEPARTMENT_ID  NUMBER(4) CONSTRAINT EMP_DEPT_FK REFERENCES DEPARTMENTS,  
  CONSTRAINT EMP_JOB_FK FOREIGN KEY (JOB_ID) REFERENCES JOBS (JOB_ID),  
  CHECK (SALARY > 0)  
);
```

Out-of-Line Constraints

```
CREATE TABLE EMPLOYEES (  
  EMPLOYEE_ID    NUMBER(6),  
  FIRST_NAME     VARCHAR2(20),  
  LAST_NAME      VARCHAR2(25) CONSTRAINT EMP_LAST_NAME_NN NOT NULL,  
  PHONE_NUMBER   VARCHAR2(20),  
  HIRE_DATE      DATE NOT NULL,  
  JOB_ID         VARCHAR2(10) NOT NULL,  
  SALARY         NUMBER(8,2),  
  COMMISSION_PCT NUMBER(2,2),  
  MANAGER_ID     NUMBER(6),  
  DEPARTMENT_ID  NUMBER(4) CONSTRAINT EMP_DEPT_FK REFERENCES DEPARTMENTS,  
  CONSTRAINT EMP_JOB_FK FOREIGN KEY (JOB_ID) REFERENCES JOBS (JOB_ID),  
  CHECK (SALARY > 0)  
);
```

Constraint Names

```
CREATE TABLE EMPLOYEES (  
  EMPLOYEE_ID    NUMBER(6),  
  FIRST_NAME     VARCHAR2(20),  
  LAST_NAME      VARCHAR2(25) CONSTRAINT EMP_LAST_NAME_NN NOT NULL,  
  PHONE_NUMBER   VARCHAR2(20),  
  HIRE_DATE      DATE NOT NULL,  
  JOB_ID         VARCHAR2(10) NOT NULL,  
  SALARY         NUMBER(8,2),  
  COMMISSION_PCT NUMBER(2,2),  
  MANAGER_ID     NUMBER(6),  
  DEPARTMENT_ID  NUMBER(4) CONSTRAINT EMP_DEPT_FK REFERENCES DEPARTMENTS,  
  CONSTRAINT EMP_JOB_FK FOREIGN KEY (JOB_ID) REFERENCES JOBS (JOB_ID),  
  CHECK (SALARY > 0)  
);
```

Constraint Names

```
CREATE TABLE EMPLOYEES (  
  EMPLOYEE_ID    NUMBER(6),  
  FIRST_NAME     VARCHAR2(20),  
  LAST_NAME      VARCHAR2(25) CONSTRAINT EMP_LAST_NAME_NN NOT NULL,  
  PHONE_NUMBER   VARCHAR2(20),  
  HIRE_DATE      DATE NOT NULL,  
  JOB_ID         VARCHAR2(10) NOT NULL,  
  SALARY         NUMBER(8,2),  
  COMMISSION_PCT NUMBER(2,2),  
  MANAGER_ID     NUMBER(6),  
  DEPARTMENT_ID NUMBER(4) CONSTRAINT EMP_DEPT_FK REFERENCES DEPARTMENTS,  
  CONSTRAINT EMP_JOB_FK FOREIGN KEY (JOB_ID) REFERENCES JOBS (JOB_ID),  
  CHECK (SALARY > 0);  
);
```



SYS_C0076043

Multiple Inline Constraints

```
ALTER TABLE EMPLOYEES ADD (  
    EMAIL VARCHAR2(25)  
        NOT NULL  
        CONSTRAINT EMP_EMAIL_UK UNIQUE  
        CONSTRAINT EMP_EMAIL_CHK CHECK (email like '%@mycompany.com')  
);
```

Multiple Inline Constraints

```
ALTER TABLE EMPLOYEES ADD (  
    EMAIL VARCHAR2(25)  
        NOT NULL  
        CONSTRAINT EMP_EMAIL_UK UNIQUE  
        CONSTRAINT EMP_EMAIL_CHK CHECK (email like '%@mycompany.com')  
);
```


Adding Inline Constraints to Existing Columns

```
ALTER TABLE EMPLOYEES MODIFY (  
    EMPLOYEE_ID CONSTRAINT EMP_EMP_ID_PK PRIMARY KEY  
);
```

Constraints with Multiple Columns

```
CREATE TABLE JOB_HISTORY (  
  EMPLOYEE_ID NUMBER(6)  
    CONSTRAINT JHIST_EMPLOYEE_NN NOT NULL  
    CONSTRAINT JHIST_EMP_FK REFERENCES EMPLOYEES,  
  START_DATE DATE  
    CONSTRAINT JHIST_START_DATE_NN NOT NULL,  
  END_DATE DATE  
    CONSTRAINT JHIST_END_DATE_NN NOT NULL,  
  JOB_ID VARCHAR2(10)  
    CONSTRAINT JHIST_JOB_NN NOT NULL  
    CONSTRAINT JHIST_JOB_FK REFERENCES JOBS,  
  DEPARTMENT_ID NUMBER(4)  
    CONSTRAINT JHIST_DEPT_FK REFERENCES DEPARTMENTS,  
  --  
  CONSTRAINT JHIST_EMP_ID_ST_DATE_PK PRIMARY KEY (EMPLOYEE_ID, START_DATE),  
  --  
  CONSTRAINT JHIST_DATE_INTERVAL CHECK (END_DATE > START_DATE)  
);
```

Out-of-Line Constraints – Column Names

```
CREATE TABLE JOB_HISTORY (  
  EMPLOYEE_ID NUMBER(6)  
    CONSTRAINT JHIST_EMPLOYEE_NN NOT NULL  
    CONSTRAINT JHIST_EMP_FK REFERENCES EMPLOYEES,  
  START_DATE DATE  
    CONSTRAINT JHIST_START_DATE_NN NOT NULL,  
  END_DATE DATE  
    CONSTRAINT JHIST_END_DATE_NN NOT NULL,  
  JOB_ID VARCHAR2(10)  
    CONSTRAINT JHIST_JOB_NN NOT NULL  
    CONSTRAINT JHIST_JOB_FK REFERENCES JOBS,  
  DEPARTMENT_ID NUMBER(4)  
    CONSTRAINT JHIST_DEPT_FK REFERENCES DEPARTMENTS,  
  --  
  CONSTRAINT JHIST_EMP_ID_ST_DATE_PK PRIMARY KEY (EMPLOYEE_ID, START_DATE),  
  --  
  CONSTRAINT JHIST_DATE_INTERVAL CHECK (END_DATE > START_DATE)  
);
```

Inline Constraints – No Column Names

```
CREATE TABLE JOB_HISTORY (  
    EMPLOYEE_ID NUMBER(6)  
        CONSTRAINT JHIST_EMPLOYEE_NN NOT NULL  
        CONSTRAINT JHIST_EMP_FK REFERENCES EMPLOYEES,  
    START_DATE DATE  
        CONSTRAINT JHIST_START_DATE_NN NOT NULL,  
    END_DATE DATE  
        CONSTRAINT JHIST_END_DATE_NN NOT NULL,  
    JOB_ID VARCHAR2(10)  
        CONSTRAINT JHIST_JOB_NN NOT NULL  
        CONSTRAINT JHIST_JOB_FK REFERENCES JOBS,  
    DEPARTMENT_ID NUMBER(4)  
        CONSTRAINT JHIST_DEPT_FK REFERENCES DEPARTMENTS,  
--  
    CONSTRAINT JHIST_EMP_ID_ST_DATE_PK PRIMARY KEY (EMPLOYEE_ID, START_DATE),  
--  
    CONSTRAINT JHIST_DATE_INTERVAL CHECK (END_DATE > START_DATE)  
);
```

Inline Check Constraints

```
ALTER TABLE EMPLOYEES ADD (  
    IS_INTERNAL CHAR(1) CHECK (IS_INTERNAL IN ('Y', 'N'))  
);
```

```
SQL> ALTER TABLE EMPLOYEES ADD (  
    2     IS_INTERNAL CHAR(1) CHECK (OTHER_COLUMN IN ('Y', 'N'))  
    3 );  
)  
*
```

```
ERROR at line 3:  
ORA-02438: Column check constraint cannot reference other columns
```

A decorative vertical bar on the left side of the slide, featuring a gradient of light red to white and several solid red circles of varying sizes.

CONSTRAINT STATES

Constraint States

VALIDATED


**NOT
VALIDATED**



Constraint States



ENABLED



DISABLED

	Enabled	Disabled
Validated		
Not Validated		

	Enabled	Disabled
Validated	 all the data is valid and will remain valid	
Not Validated		

	Enabled	Disabled
Validated	 all the data is valid and will remain valid	
Not Validated		 existing and future data may violate the rules

	Enabled	Disabled
Validated	 all the data is valid and will remain valid	
Not Validated	new DMLs cannot violate the rules, but existing data can	 existing and future data may violate the rules

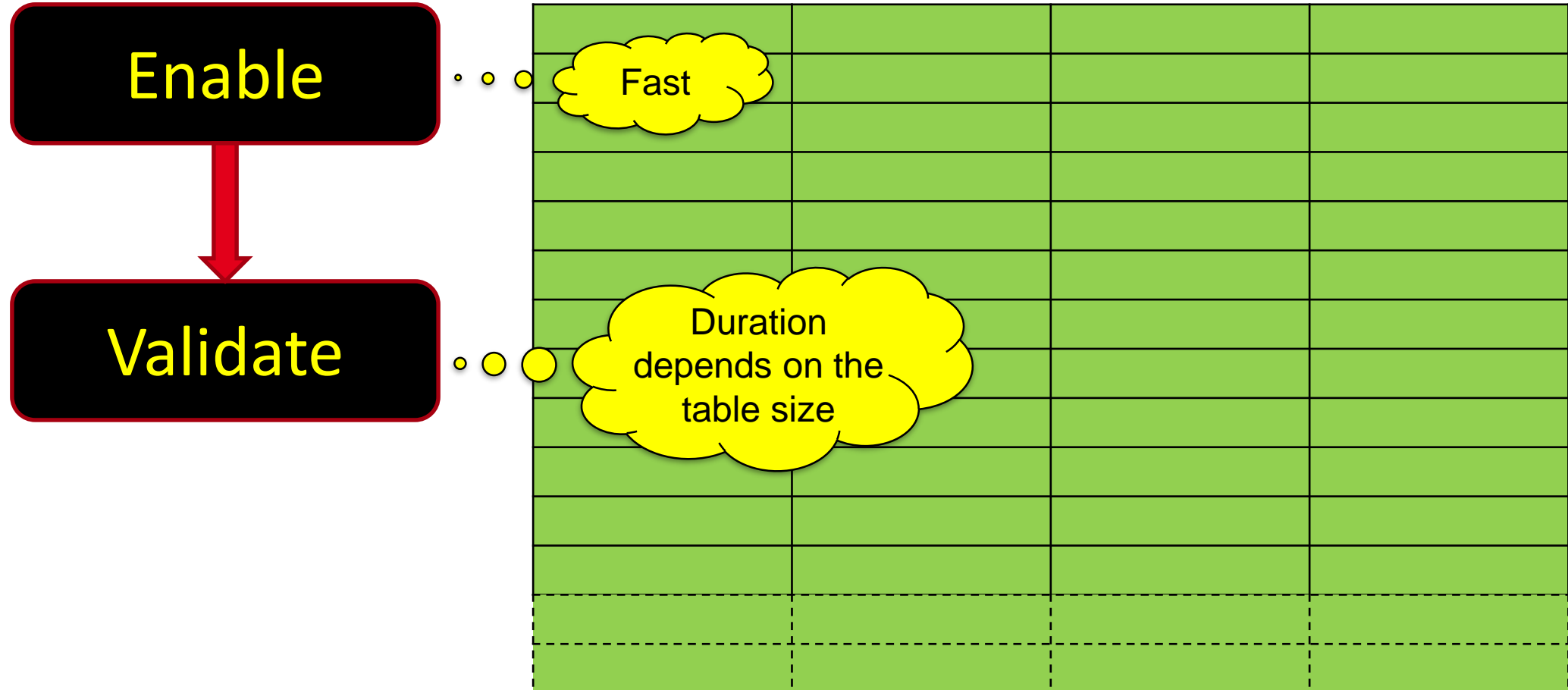
	Enabled	Disabled
Validated	 all the data is valid and will remain valid	?
Not Validated	new DMLs cannot violate the rules, but existing data can	 existing and future data may violate the rules

A decorative vertical bar on the left side of the slide, featuring a gradient from light pink to white, with several red circles of varying sizes scattered along it.

CONSTRAINT CREATION PROCESS

Constraint Creation Process

```
ALTER TABLE T ADD CONSTRAINT ... [ENABLE VALIDATE]
```



A decorative vertical bar on the left side of the slide, featuring a light red background with several vertical lines of varying thickness and a cluster of five solid red circles of different sizes.

ADDING A COLUMN WITH A CHECK CONSTRAINT

Adding a Column with a Check Constraint

```
ALTER TABLE T ADD (  
  B NUMBER CONSTRAINT B_CHK CHECK (B>0)  
);
```

Inline check constraints
for new columns are
marked as **VALIDATED**
without actually performing
the validation phase

```
ALTER TABLE T ADD (  
  C NUMBER , CONSTRAINT C_CHK CHECK (C>0)  
);
```

No optimization is done for
**out-of-line check
constraints**

OPTIMIZATION TURNS INTO A BUG



@bug

How Check Conditions are Evaluated

```
ALTER TABLE TBL ADD (  
    Y NUMBER CHECK (Y > 0)  
);
```

- The condition is Boolean
- Hence can be evaluated to one of two values:
 - TRUE
 - FALSE

How Check Conditions are Evaluated

```
ALTER TABLE TBL ADD (  
    Y NUMBER CHECK (Y > 0)  
);
```

- The condition is Boolean
- Hence can be evaluated to one of ~~two~~ ^{three} values:
 - TRUE
 - FALSE
 - NULL
- A record is considered VALID if the check constraint condition is **not evaluated to FALSE**

Y	Y > 0
5	TRUE
-5	FALSE
NULL	NULL

Y	Y IN ('A','B','C')
B	TRUE
D	FALSE
NULL	NULL

Y	Y = TRUNC(Y)
3/12/2018 00:00:00	TRUE
3/12/2018 16:15:00	FALSE
NULL	NULL

Y	Y IS NOT NULL
5	TRUE
-5	TRUE
NULL	FALSE

Y	NVL(Y,0) > 0
5	TRUE
-5	FALSE
NULL	FALSE

```
ALTER TABLE TBL ADD (  
  Y NUMBER CHECK (Y IS NOT NULL)  
)
```

```
ALTER TABLE TBL ADD (  
  Y NUMBER NOT NULL  
)
```


A decorative vertical bar on the left side of the slide, featuring a gradient from light pink to dark red. It contains several red circles of varying sizes, with the largest one at the top left and smaller ones scattered below and to the right.

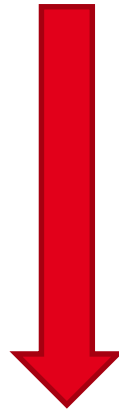
NOT NULL CONSTRAINTS

Adding a Column with a NOT NULL Constraint

```
ALTER TABLE TBL ADD (  
  Y NUMBER NOT NULL  
)
```

```
select /*+ full(P) noparallel(P) */ 1 from "DEMO2"."TBL" P where rownum = 1
```

1 row selected



no rows selected

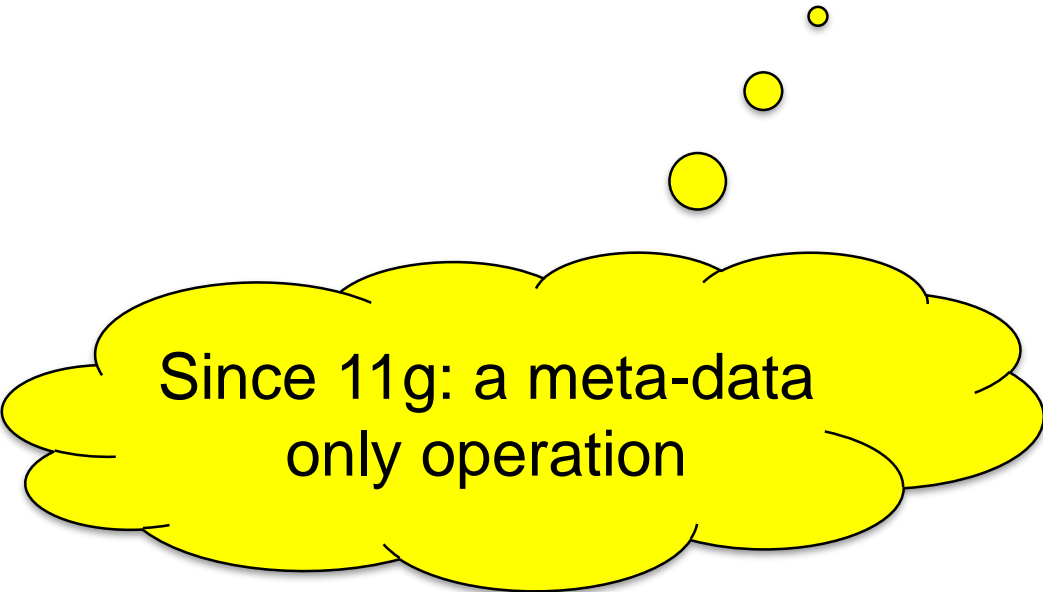


Table altered.

```
ORA-01758: table must be empty to add mandatory (NOT NULL) column
```

Adding a Column with a NOT NULL Constraint + Default

```
ALTER TABLE TBL ADD (  
  Y NUMBER DEFAULT 42 NOT NULL  
)
```



Since 11g: a meta-data
only operation

A decorative vertical bar on the left side of the slide, featuring a gradient from light pink to white, with several red circles of varying sizes and a thin red line on the right edge.

NOT NULL vs. CHECK

NOT NULL vs. CHECK

```
ALTER TABLE TBL ADD (  
  Y NUMBER CHECK (Y IS NOT NULL)  
)
```


?

=

```
ALTER TABLE TBL ADD (  
  Y NUMBER NOT NULL  
)
```

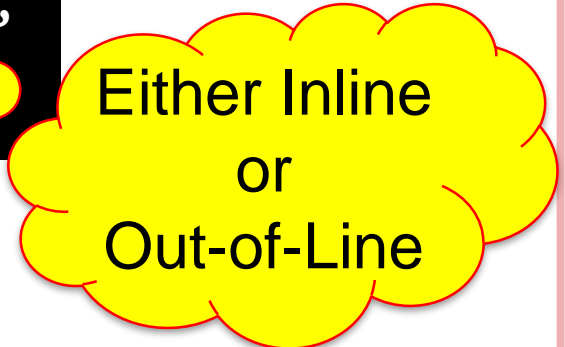
NOT NULL vs. CHECK - #1

```
CREATE TABLE T3 (  
  ID          NUMBER CONSTRAINT T3_PK          PRIMARY KEY,  
  COL_NN     NUMBER CONSTRAINT T3_COL_NN     NOT NULL,  
  COL_CHK    NUMBER CONSTRAINT T3_COL_CHK    CHECK (COL_CHK IS NOT NULL),  
  DETAILS    VARCHAR2(1000)  
);
```



NOT NULL vs. CHECK - #1

```
CREATE TABLE T3 (  
  ID          NUMBER CONSTRAINT T3_PK          PRIMARY KEY,  
  COL_NN     NUMBER CONSTRAINT T3_COL_NN     NOT NULL,  
  COL_CHK   NUMBER CONSTRAINT T3_COL_CHK   CHECK (COL_CHK IS NOT NULL),  
  DETAILS   VARCHAR2(1000)  
);
```



Either Inline
or
Out-of-Line

NOT NULL vs. CHECK - #2

```
SQL> SELECT CONSTRAINT_NAME,
2      CONSTRAINT_TYPE,
3      SEARCH_CONDITION,
4      STATUS,
5      VALIDATED
6 FROM USER_CONSTRAINTS
7 WHERE TABLE_NAME = 'T3';
```

CONSTRAINT NAME	CONST TYPE	CONDITION	STATUS	VALIDATED
T3_COL_NN	C	"COL_NN" IS NOT NULL	ENABLED	VALIDATED
T3_COL_CHK	C	COL_CHK IS NOT NULL	ENABLED	VALIDATED
T3_PK	P		ENABLED	VALIDATED

Implicit Check Constraint

Explicit Check Constraint

NOT NULL vs. CHECK - #3

```
SQL> SELECT COLUMN_NAME,  
2         NULLABLE  
3 FROM USER_TAB_COLUMNS  
4 WHERE TABLE_NAME='T3';
```

COLUMN_NAME	NULLABLE
-----	-----
ID	N
COL_NN	N
COL_CHK	Y
DETAILS	Y

NOT NULL vs. CHECK - #3

```
SQL> SELECT COLUMN_NAME,  
2         NULLABLE  
3 FROM USER_TAB_COLUMNS  
4 WHERE TABLE_NAME='T3';
```

COLUMN_NAME	NULLABLE
-------------	----------

ID	N
----	---

COL_NN	N
--------	----------

COL_CHK	Y
---------	---

DETAILS	Y
---------	---

```
CREATE TABLE T3 (  
  ID          NUMBER CONSTRAINT T3_PK          PRIMARY KEY,  
  COL_NN     NUMBER CONSTRAINT T3_COL_NN     NOT NULL,  
  COL_CHK    NUMBER CONSTRAINT T3_COL_CHK    CHECK (COL_CHK IS NOT NULL),  
  DETAILS    VARCHAR2(1000)  
);
```

NOT NULL vs. CHECK - #3

```
SQL> SELECT COLUMN_NAME,  
2         NULLABLE  
3 FROM USER_TAB_COLUMNS  
4 WHERE TABLE_NAME='T3';
```

COLUMN_NAME	NULLABLE
-------------	----------

ID	N
----	---

COL_NN	N
--------	---

COL_CHK	Y
---------	---

DETAILS	Y
---------	---

```
CREATE TABLE T3 (  
  ID          NUMBER CONSTRAINT T3_PK          PRIMARY KEY,  
  COL_NN     NUMBER CONSTRAINT T3_COL_NN     NOT NULL,  
  COL_CHK    NUMBER CONSTRAINT T3_COL_CHK    CHECK (COL_CHK IS NOT NULL),  
  DETAILS    VARCHAR2(1000)  
);
```

NOT NULL vs. CHECK - #3

```
SQL> SELECT COLUMN_NAME,  
2         NULLABLE  
3 FROM USER_TAB_COLUMNS  
4 WHERE TABLE_NAME='T3';
```

```
COLUMN_NAME  NULLABLE  
-----  -
```

ID	N
COL_NN	N
COL_CHK	Y
DETAILS	Y

```
CREATE TABLE T3 (  
ID          NUMBER CONSTRAINT T3_PK          PRIMARY KEY,  
COL_NN     NUMBER CONSTRAINT T3_COL_NN     NOT NULL,  
COL_CHK    NUMBER CONSTRAINT T3_COL_CHK    CHECK (COL_CHK IS NOT NULL),  
DETAILS    VARCHAR2(1000)  
);
```

NOT NULL vs. CHECK - #4

```
SQL> DESC T3
```

Name	Null?	Type
-----	-----	-----
ID	NOT NULL	NUMBER
COL_NN	NOT NULL	NUMBER
COL_CHK		NUMBER
DETAILS		VARCHAR2(1000)

NOT NULL vs. CHECK - #4

```
SQL> DESC T3
```

Name	Null?	Type
-----	-----	-----
ID	NOT NULL	NUMBER
COL_NN	NOT NULL	NUMBER
COL_CHK		NUMBER
DETAILS		VARCHAR2(1000)

```
SQL> SELECT COLUMN_NAME,  
2          NULLABLE  
3 FROM USER_TAB_COLUMNS  
4 WHERE TABLE_NAME='T3';
```

COLUMN_NAME	NULLABLE
-----	-----
ID	N
COL_NN	N
COL_CHK	Y
DETAILS	Y

NOT NULL vs. CHECK - #4

```
SQL> DESC T3
```

Name	Null?	Type
-----	-----	-----
ID	NOT NULL	NUMBER
COL_NN	NOT NULL	NUMBER
COL_CHK		NUMBER
DETAILS		VARCHAR2(1000)

```
SQL> SELECT COLUMN_NAME,  
2          NULLABLE  
3 FROM USER_TAB_COLUMNS  
4 WHERE TABLE_NAME='T3';
```

COLUMN_NAME	NULLABLE
-----	-----
ID	N
COL_NN	N
COL_CHK	Y
DETAILS	Y

NOT NULL vs. CHECK - #5

```
SQL> INSERT INTO T3 (ID, COL_NN, COL_CHK)
  2  VALUES (1, 1, NULL);
```

```
INSERT INTO T3 (ID, COL_NN, COL_CHK)
```

```
*
```

```
ERROR at line 1:
```

```
ORA-02290: check constraint (DEM02.T3_COL_CHK) violated
```

```
SQL> INSERT INTO T3 (ID, COL_NN, COL_CHK)
  2  VALUES (1, NULL, 1);
```

```
VALUES (1, NULL, 1)
```

```
*
```

```
ERROR at line 2:
```

```
ORA-01400: cannot insert NULL into ("DEM02"."T3"."COL_NN")
```


NOT NULL vs. CHECK - #5

```
SQL> INSERT INTO T3 (ID, COL_NN, COL_CHK)
2 VALUES (1, 1, NULL);
```

```
INSERT INTO T3 (ID, COL_NN, COL_CHK)
```

```
*
```

```
ERROR at line 1:
```

```
ORA-02290: check constraint (DEMO2.T3_COL_CHK) violated
```

```
SQL> INSERT INTO T3 (ID, COL_NN, COL_CHK)
2 VALUES (1, NULL, 1);
```

```
VALUES (1, NULL, 1)
```

```
*
```

```
ERROR at line 2:
```

```
ORA-01400: cannot insert NULL into ("DEMO2"."T3"."COL_NN")
```

NOT NULL vs. CHECK - #6

```
SELECT * FROM T3
WHERE COL_NN IS NULL;
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	1013	0 (0)	
* 1	FILTER					
2	TABLE ACCESS FULL	T3	1000	989K	44 (0)	00:00:01

Predicate Information (identified by operation id):

1 - filter(NULL IS NOT NULL)

NOT NULL vs. CHECK - #6

```
SELECT * FROM T3
WHERE COL_CHK IS NULL;
```

```
-----
| Id | Operation | Name | Rows | Bytes | Cost (%CPU) | Time |
-----
| 0 | SELECT STATEMENT | | 1 | 1013 | 44 (0) | 00:00:01 |
|* 1 | TABLE ACCESS FULL | T3 | 1 | 1013 | 44 (0) | 00:00:01 |
-----
```

Predicate Information (identified by operation id):

```
-----
```

```
1 - filter("COL_CHK" IS NULL)
```

NOT NULL vs. CHECK - #6

```
CREATE INDEX T3_IDX_NN ON T3(COL_NN);
```

```
SELECT COL_NN FROM T3;
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1000	4000	3 (0)	00:00:01
1	INDEX FAST FULL SCAN	T3_IDX_NN	1000	4000	3 (0)	00:00:01

NOT NULL vs. CHECK - #6

```
CREATE INDEX T3_IDX_CHK ON T3(COL_CHK);
```

```
SELECT COL_CHK FROM T3;
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1000	4000	44 (0)	00:00:01
1	TABLE ACCESS FULL	T3	1000	4000	44 (0)	00:00:01

NOT NULL vs. CHECK - #6

```
ALTER TABLE T3 ADD CONSTRAINT T3_CHK2
CHECK (COL_CHK <= 1000);
```

```
SELECT *
FROM T3
WHERE COL_CHK = 2000;
```

The Optimizer Can't?
Or Won't?

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	1013	0 (0)	
* 1	FILTER					
2	TABLE ACCESS BY INDEX ROWID BATCHED	T3	1	1013	2 (0)	00:00:01
* 3	INDEX RANGE SCAN	T3_IDX_CHK	1		1 (0)	00:00:01

Predicate Information (identified by operation id):

- 1 - filter(NULL IS NOT NULL)
- 3 - access("COL_CHK"=2000)

NOT NULL vs. CHECK - #7

```
CREATE VIEW V3 AS SELECT * FROM T3;
```

```
CREATE PROCEDURE P3 AS  
  L_CHK V3.COL_CHK%TYPE;  
  L_NN  V3.COL_NN%TYPE;  
BEGIN  
  NULL;  
END P3;  
/
```

```
SELECT OBJECT_NAME,OBJECT_TYPE,STATUS  
FROM USER_OBJECTS  
WHERE OBJECT_NAME LIKE '_3';
```

OBJECT_NAME	OBJECT_TYPE	STATUS
P3	PROCEDURE	VALID
T3	TABLE	VALID
V3	VIEW	VALID

NOT NULL vs. CHECK - #7

```
ALTER TABLE T3 DROP CONSTRAINT T3_COL_NN;
```

```
SELECT OBJECT_NAME,OBJECT_TYPE,STATUS  
FROM USER_OBJECTS  
WHERE OBJECT_NAME LIKE '_3';
```

OBJECT_NAME	OBJECT_TYPE	STATUS
P3	PROCEDURE	INVALID
T3	TABLE	VALID
V3	VIEW	INVALID

```
CREATE PROCEDURE P3 AS  
  L_CHK V3.COL_CHK%TYPE;  
  L_NN  V3.COL_NN%TYPE;  
BEGIN  
  NULL;  
END P3;  
/
```


NOT NULL vs. CHECK - #7

```
EXEC DBMS_UTILITY.COMPILE_SCHEMA(USER)
```

```
SELECT OBJECT_NAME,OBJECT_TYPE,STATUS  
FROM USER_OBJECTS  
WHERE OBJECT_NAME LIKE '_3';
```

OBJECT_NAME	OBJECT_TYPE	STATUS
P3	PROCEDURE	VALID
T3	TABLE	VALID
V3	VIEW	VALID

NOT NULL vs. CHECK - #7

```
ALTER TABLE T3 DROP CONSTRAINT T3_COL_CHK;
```

```
SELECT OBJECT_NAME,OBJECT_TYPE,STATUS  
FROM USER_OBJECTS  
WHERE OBJECT_NAME LIKE '_3';
```

OBJECT_NAME	OBJECT_TYPE	STATUS
P3	PROCEDURE	VALID
T3	TABLE	VALID
V3	VIEW	VALID

```
CREATE PROCEDURE P3 AS  
  L_CHK V3.COL_CHK%TYPE;  
  L_NN  V3.COL_NN%TYPE;  
BEGIN  
  NULL;  
END P3;  
/
```



ADDING A COLUMN WITH A FOREIGN KEY CONSTRAINT

Adding a Column with a Foreign Key Constraint

T1

X	Y	CAT_ID

CATEGORIES

ID	NAME



@colfk1

Adding a Column with a Foreign Key Constraint

```
SELECT /*+ all_rows ordered dynamic_sampling(2) */
  A.ROWID, :1, :2, :3
FROM   "DEMO2"."T1"          A,
       "DEMO2"."CATEGORIES" B
WHERE  ("A"."CAT_ID" IS NOT NULL)
AND    ("B"."ID" (+) = "A"."CAT_ID")
AND    ("B"."ID" IS NULL)
```

Adding a Column with a Foreign Key Constraint

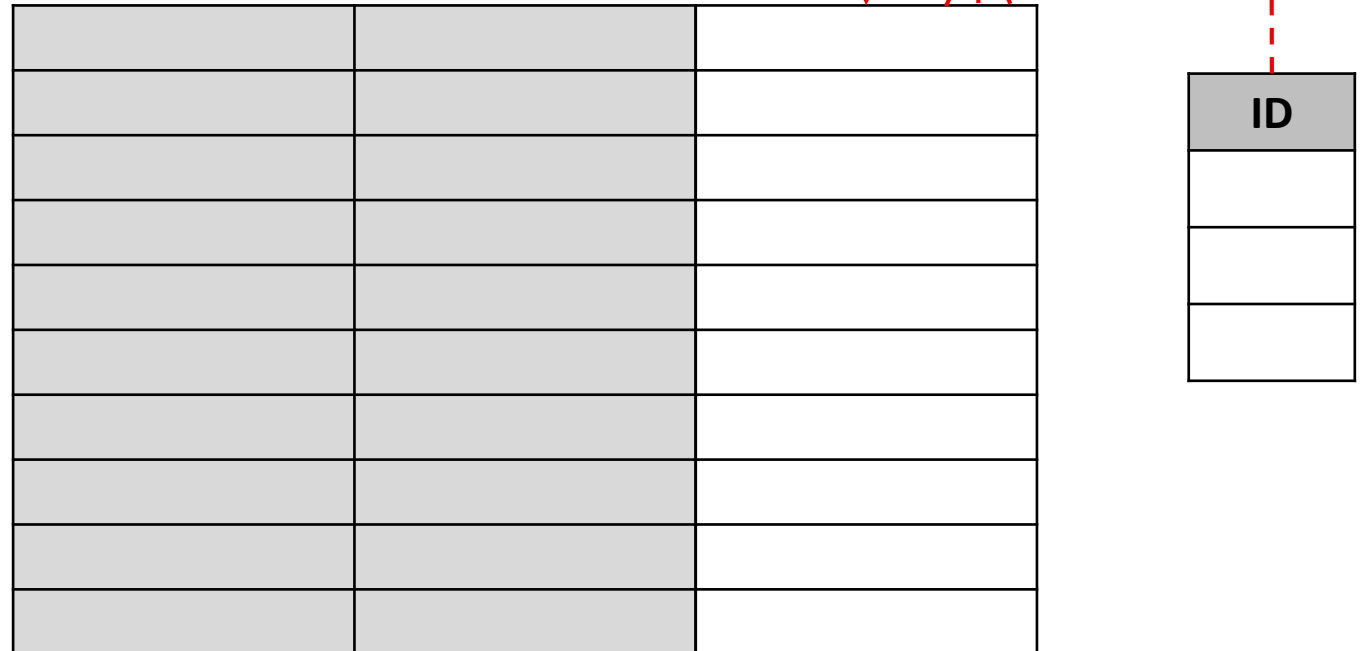
```
ALTER TABLE T1 ADD (  
  CAT_ID NUMBER CONSTRAINT FK_T1_CAT REFERENCES CATEGORIES (ID)  
);
```

Add a Column

Add a Foreign Key
Constraint on that
Column

Fast

```
ALTER TABLE T1 ADD (CAT_ID NUMBER)
```



@colfk2

Adding a Column with a Foreign Key Constraint

```
ALTER TABLE T1 ADD (  
  CAT_ID NUMBER CONSTRAINT FK_T1_CAT REFERENCES CATEGORIES (ID)  
);
```

```
ALTER TABLE T1 ADD (  
  CAT_ID NUMBER,  
  CONSTRAINT FK_T1_CAT FOREIGN KEY (CAT_ID) REFERENCES CATEGORIES (ID)  
);
```

Inline **and** out-of-line **foreign key constraints** for new columns are marked as **VALIDATED** without actually performing the validation phase

A decorative vertical bar on the left side of the slide, featuring a gradient from light pink to white and several red circles of varying sizes.

So, WHAT'S THE CATCH?

DDL

Add a Column

Add a Check/FK Constraint
on that Column

Fast

Slow (for
big tables)

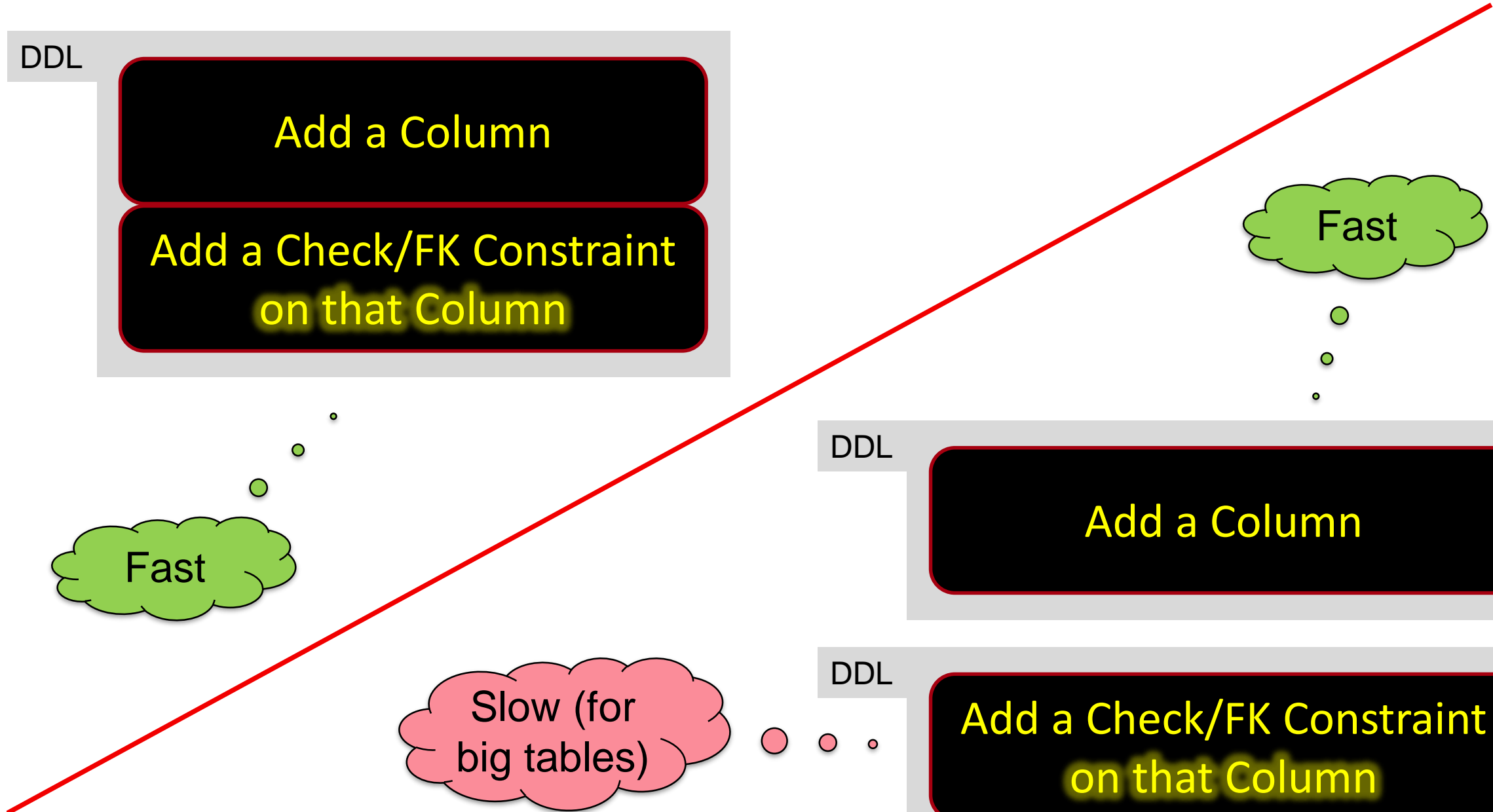
Fast

DDL

Add a Column

DDL

Add a Check/FK Constraint
on that Column



DDL

Add a Column

Add a Check/FK Constraint
on that Column

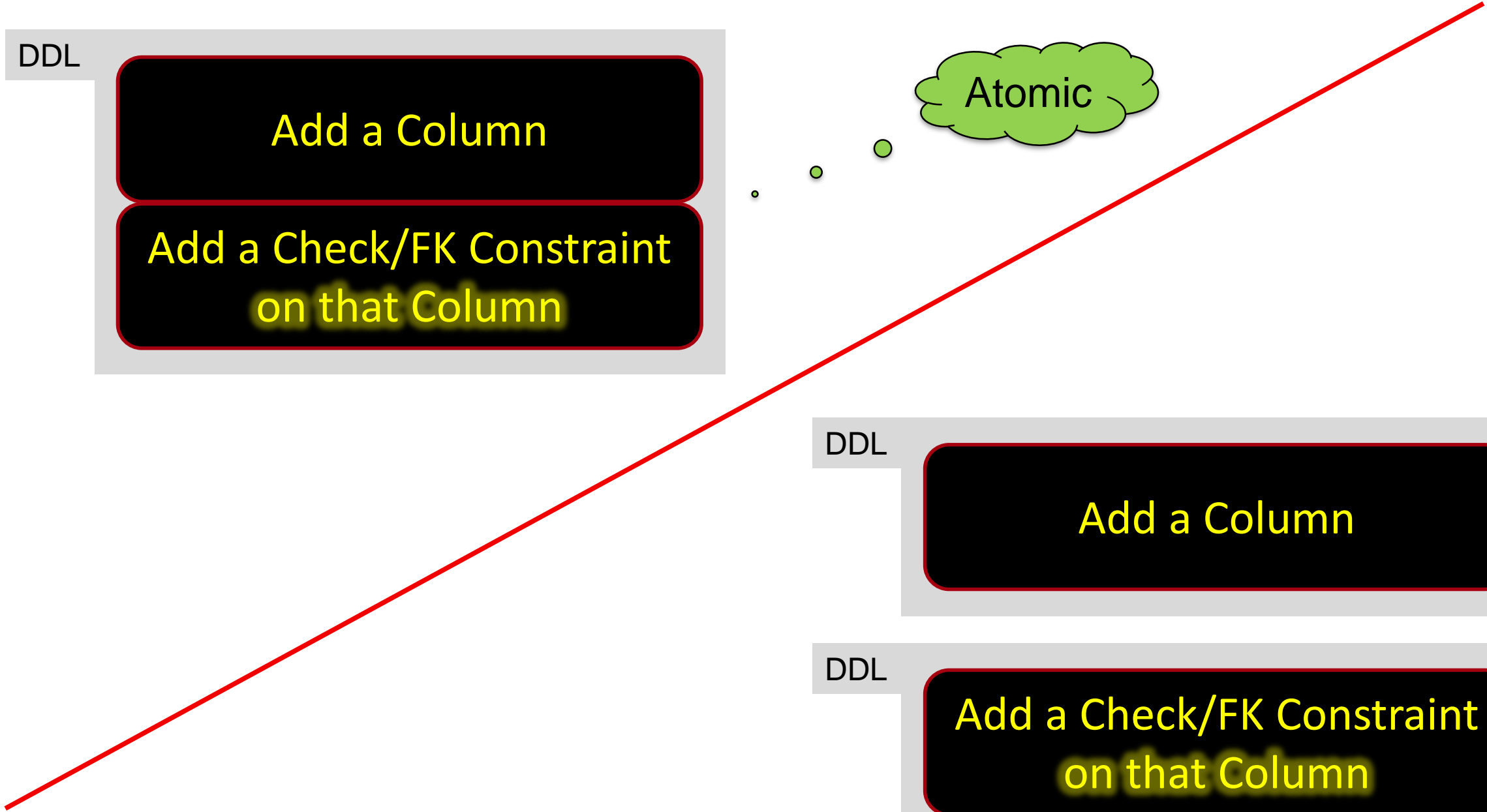
Atomic

DDL

Add a Column

DDL

Add a Check/FK Constraint
on that Column



DDL

Add a Column

Add a Check/FK Constraint
on that Column

Offline

Online

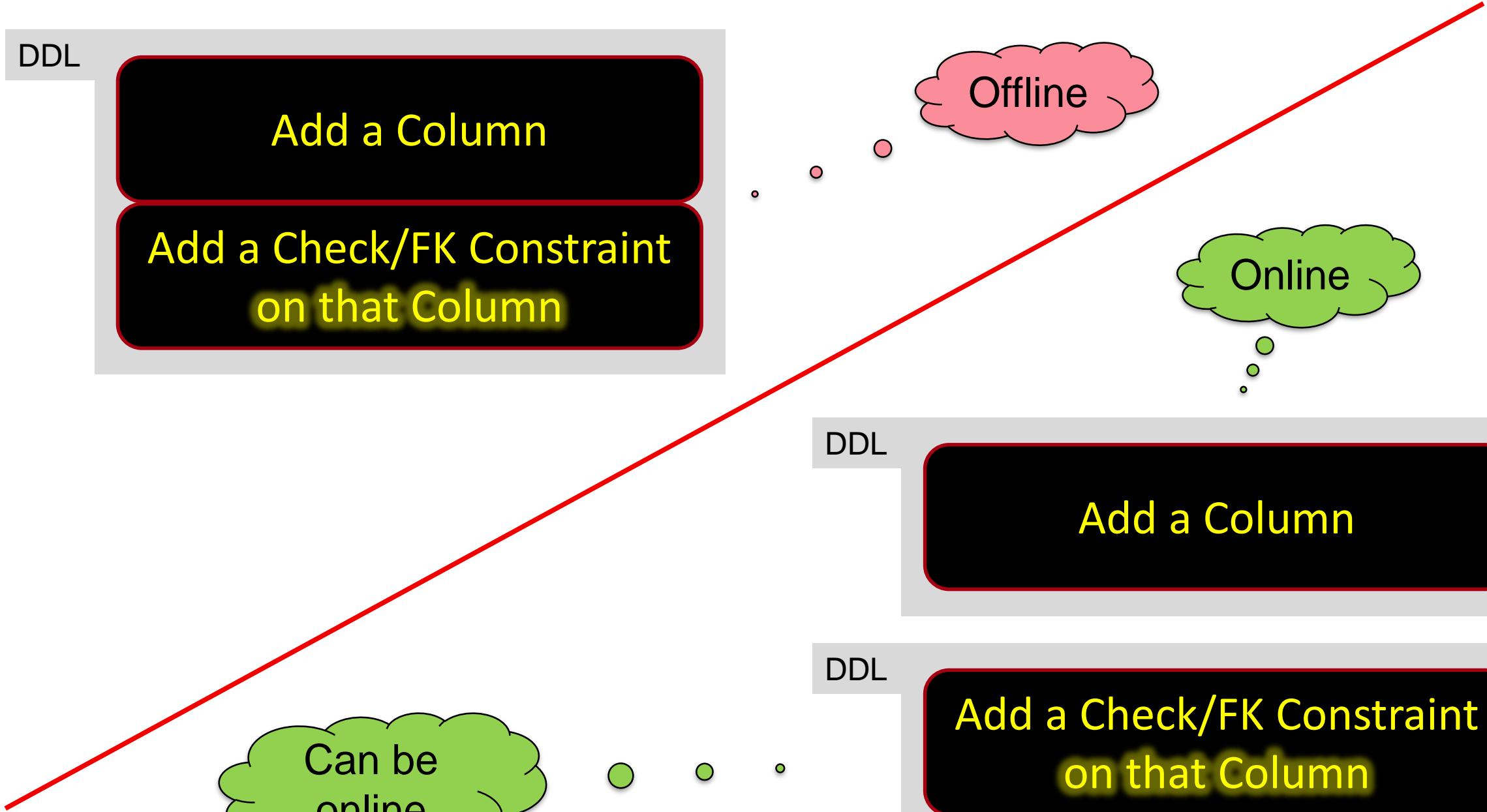
Can be
online

DDL

Add a Column

DDL

Add a Check/FK Constraint
on that Column



Offline
DDL Operations

Online
DDL Operations

Offline DDL Operations	Online DDL Operations
Acquire highly restrictive table locks	Acquire modest table locks

Offline DDL Operations	Online DDL Operations
Acquire highly restrictive table locks	Acquire modest table locks
Get ORA-54 due to active transactions	Wait for active transactions to end

Offline DDL Operations	Online DDL Operations
Acquire highly restrictive table locks	Acquire modest table locks
Get ORA-54 due to active transactions	Wait for active transactions to end
Block new DML statements	Do not block new DML statements

Offline

DDL

Add a Column

Add a Check/FK Constraint
on that Column

@offline

Online

DDL

Add a Column

DDL

*Can be
online*

Add a Check/FK Constraint
on that Column

@addcol

**Can be
online**

DDL

**Add a Check/FK Constraint
on that Column**



DDL

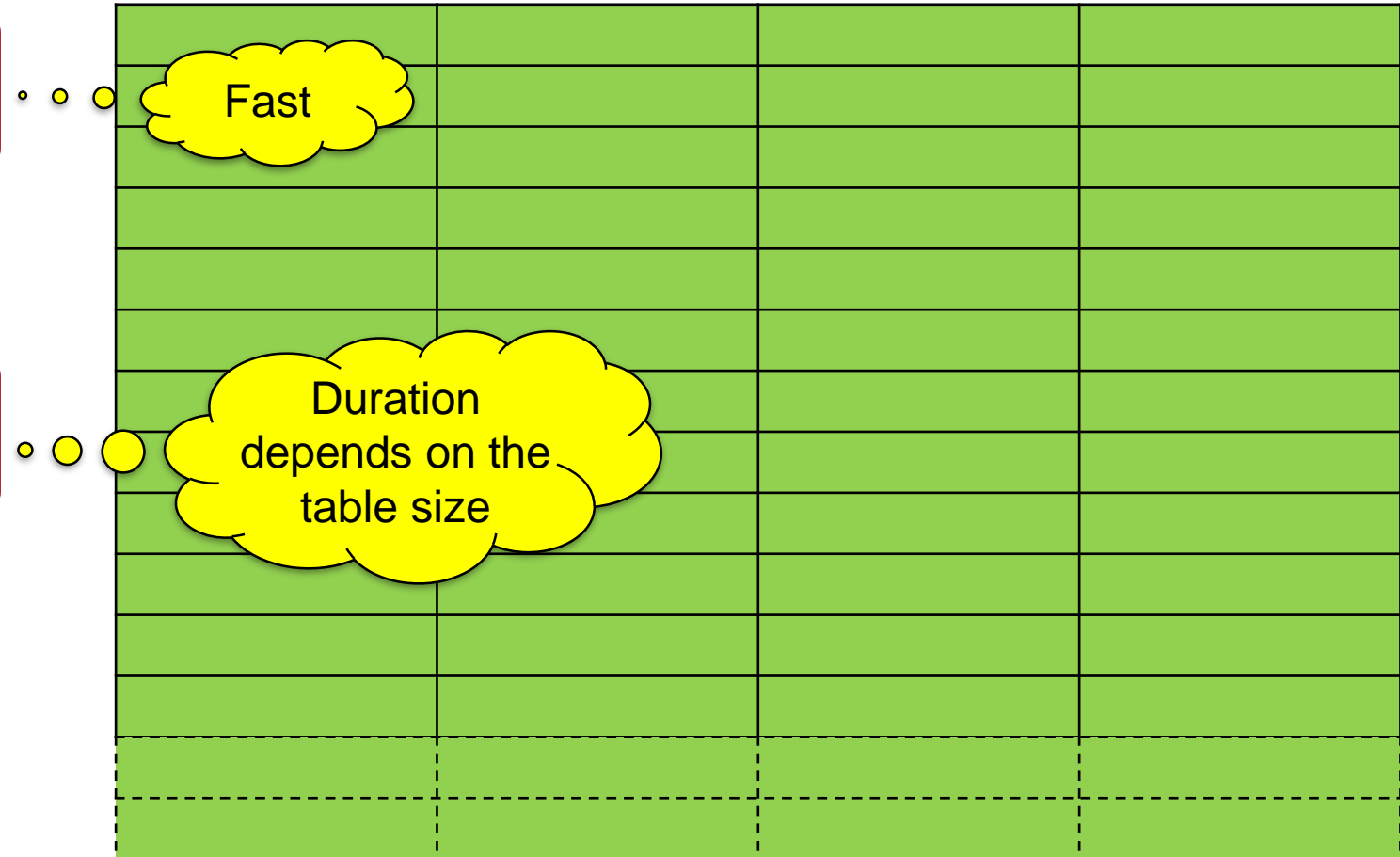
**Add a Check/FK Constraint
with `ENABLE VALIDATE`**

```
ALTER TABLE T ADD CONSTRAINT ... [ENABLE VALIDATE]
```

Enable



Validate



But not atomic...

Online

DDL

Add a Check/FK Constraint
with **ENABLE VALIDATE**

Online

@addcon



ADDING A COLUMN WITH A UNIQUE CONSTRAINT

Adding a Column with a Unique Constraint

```
PARSING IN CURSOR #2745189968080 len=67 dep=1 uid=111 oct=9 lid=111 tim=374932428211 hv=3900433436 ad='7ff9d3d5f630'  
sqlid='b7j5hhmn7rt0w'  
CREATE UNIQUE INDEX "DEM02"."T_Z_UK" on "DEM02"."T"("Z") NOPARALLEL  
END OF STMT  
.  
.  
.  
EXEC #2745189968080:c=14875000,e=38682857,p=1428572,cr=1429251,cu=71,mis=0,r=0,dep=1,og=1,plh=4149467226,tim=374971111124  
STAT #2745189968080 id=1 cnt=1 pid=0 pos=1 obj=0 op='INDEX BUILD UNIQUE T_Z_UK (cr=1428612 pr=1428572 pw=0 str=1 time=38643350  
us)'  
STAT #2745189968080 id=2 cnt=10000000 pid=1 pos=1 obj=0 op='SORT CREATE INDEX (cr=1428612 pr=1428572 pw=0 str=1 time=38410492 us)'  
STAT #2745189968080 id=3 cnt=10000000 pid=2 pos=1 obj=83618 op='TABLE ACCESS FULL T  
(cr=1428612 pr=1428572 pw=0 str=1 time=2374128 us cost=388405 size=130000000  
card=10000000)'  
CLOSE #2745189968080:c=0,e=2,dep=1,type=0,tim=3749711111289
```


An Idea for Oracle

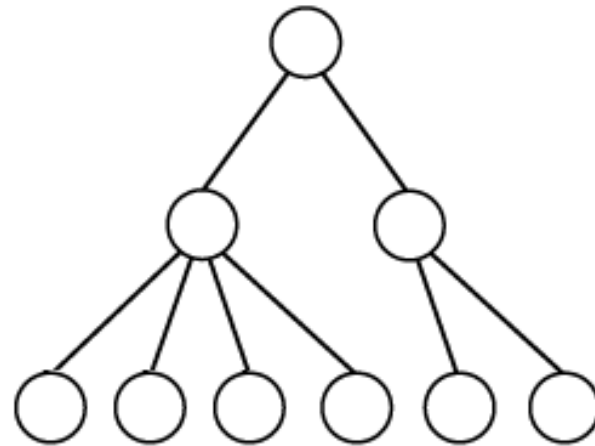
- Create an empty index in this case (without scanning all the table rows)
 - It will save time
 - It will improve availability
 - Especially for Standard Edition
- <https://community.oracle.com/ideas/17672>



A decorative vertical bar on the left side of the slide, featuring a gradient from light pink to white, with several red circles of varying sizes scattered along it.

ADDING A UNIQUE CONSTRAINT

```
ALTER TABLE T ADD CONSTRAINT T_UK UNIQUE (COL);
```



```
CREATE UNIQUE INDEX T_UK ON T (COL) ONLINE;
```



Requires
Enterprise
Edition



Offline

```
ALTER TABLE T  
ADD CONSTRAINT T_UK UNIQUE (COL)  
USING INDEX T_UK;
```

```
CREATE UNIQUE INDEX T_UK ON T (COL) ONLINE;
```

```
ALTER TABLE T  
ADD CONSTRAINT T_UK UNIQUE (COL)  
USING INDEX T_UK  
ENABLE NOVALIDATE;
```

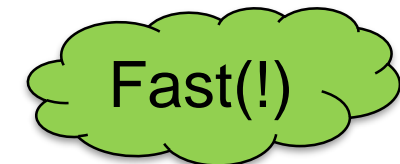
```
ALTER TABLE T  
ENABLE VALIDATE CONSTRAINT T_UK;
```

Online

Fast

Online

Fast(!)



```
PARSING IN CURSOR #2610213170320 len=101 dep=1 uid=0 oct=3 lid=0
tim=1365857610682 hv=439303422 ad='7ff9e7c80790' sqlid='cqj35uhd2yg7y'
```

```
select /*+ all_rows ordered dynamic_sampling(2) */
  A.rowid, :1, :2, :3
from "DEMO2"."T1" A
where 1=0
```

```
END OF STMT
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT				1 (100)	
* 1	FILTER					
2	TABLE ACCESS FULL	T1	1000K	11M	38957 (1)	00:00:02

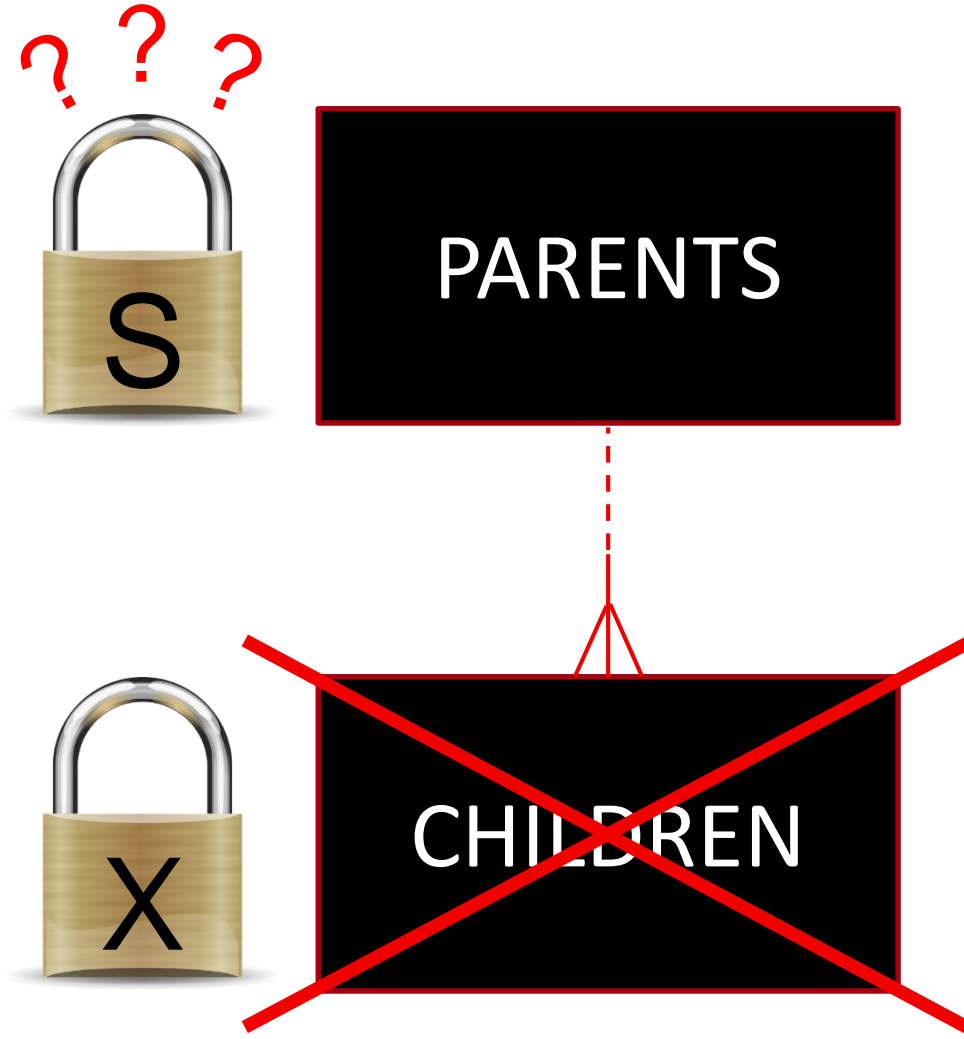
```
Predicate Information (identified by operation id):
```

```
-----
1 - filter(NULL IS NOT NULL)
```

A decorative vertical bar on the left side of the slide, featuring a gradient from light pink to dark red and several solid red circles of varying sizes.

FOREIGN KEYS AND LOCKING ISSUES

Dropping a Child Table



@dropchd

A decorative vertical bar on the left side of the page, featuring a gradient from light pink to white, with several red circles of varying sizes scattered along it.

DISABLED AND VALIDATED

THANK YOU 😊

Oren Nakdimon

www.db-oriented.com

✉ oren@db-oriented.com

☎ +972-54-4393763

🐦 @DBoriented