Application security (behind Oracle roles and profiles)

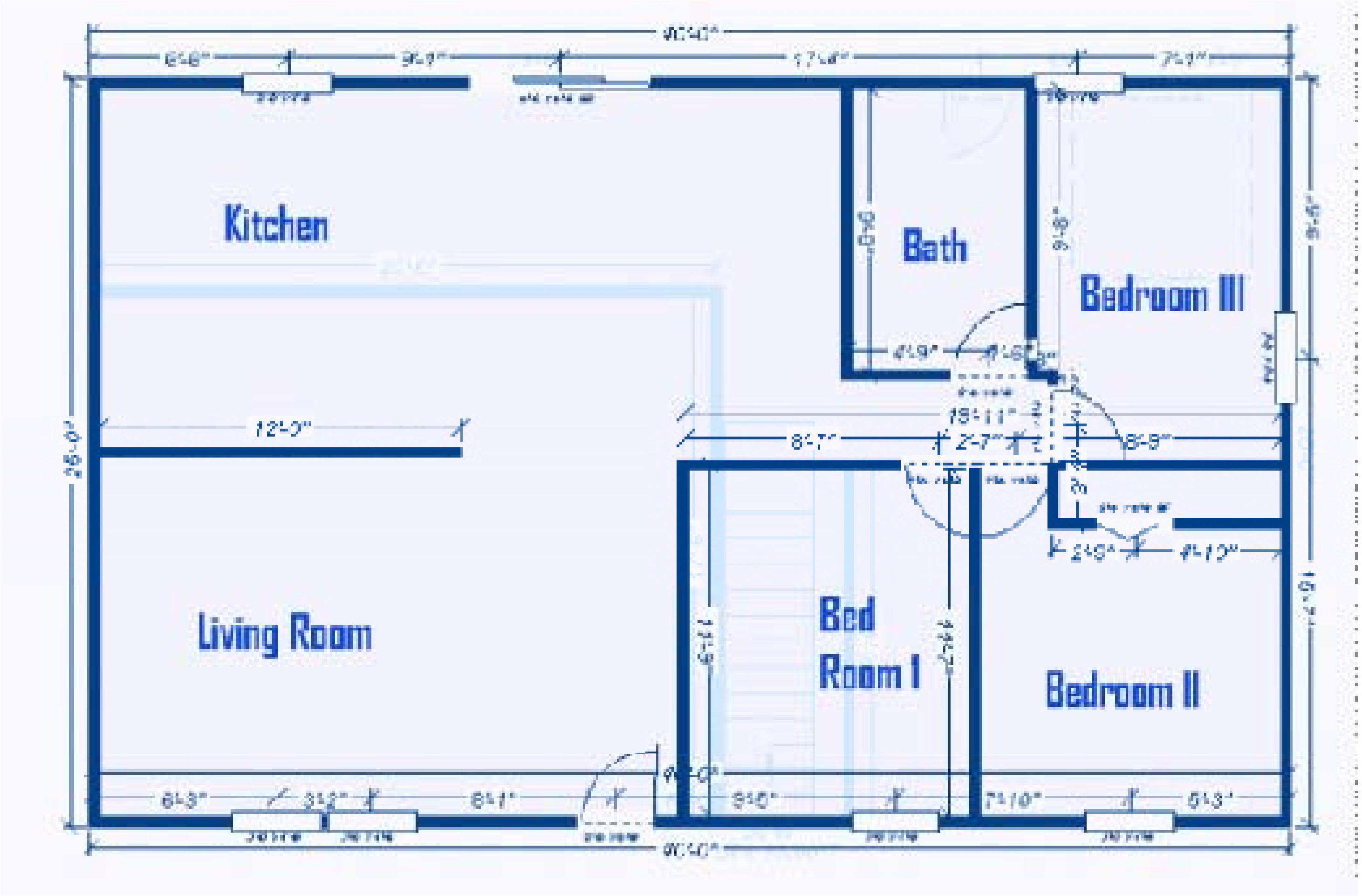
Miguel Anjo

8th July 2008

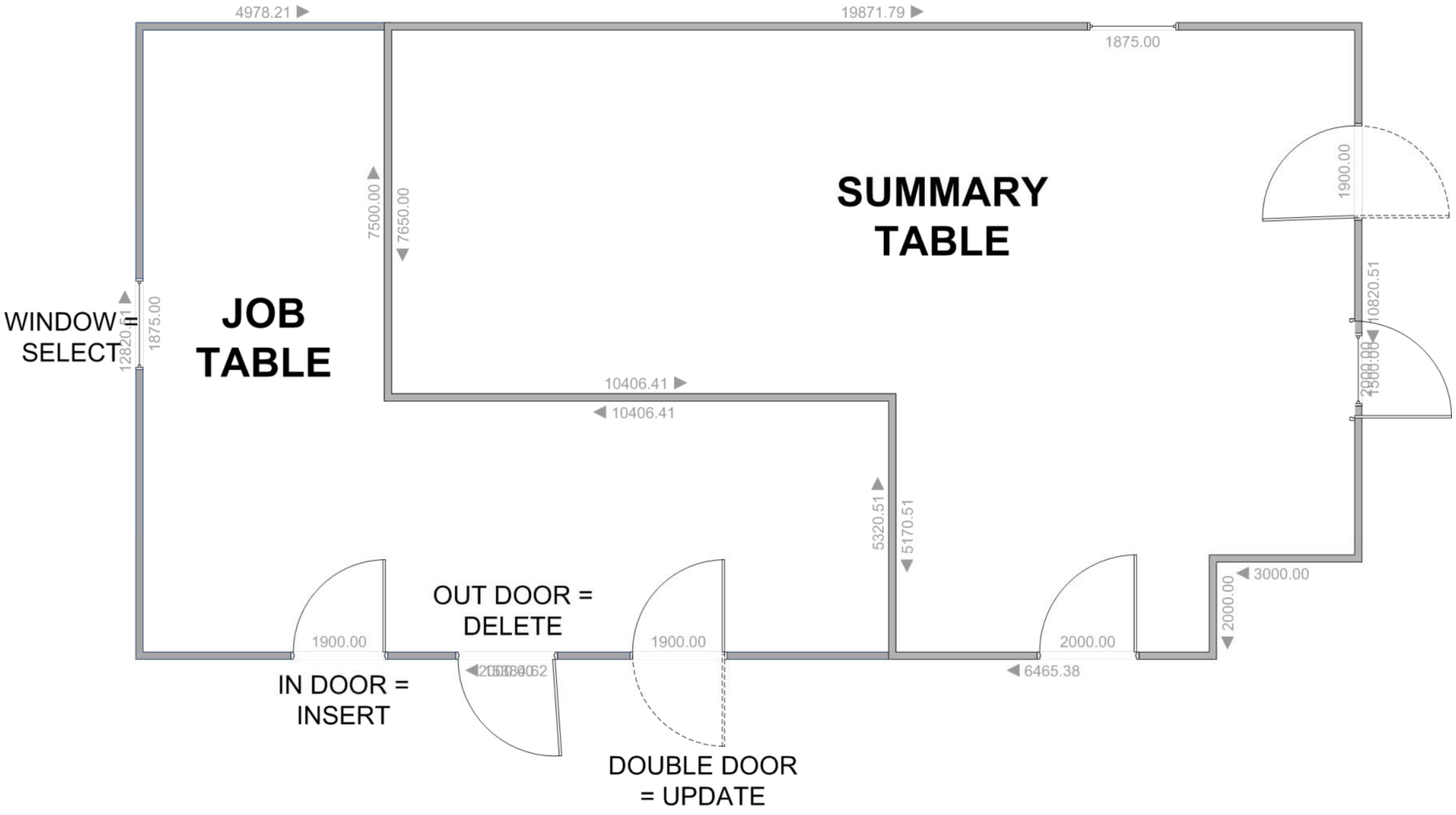
Database Developers’ Workshop

Abloy key

Floor plan


Kitchen
Bath
Bedroom III
12'-0"
Living Room
Bed Room 1
Bedroom II

Floor plan - 2



JOB TABLE

SUMMARY TABLE

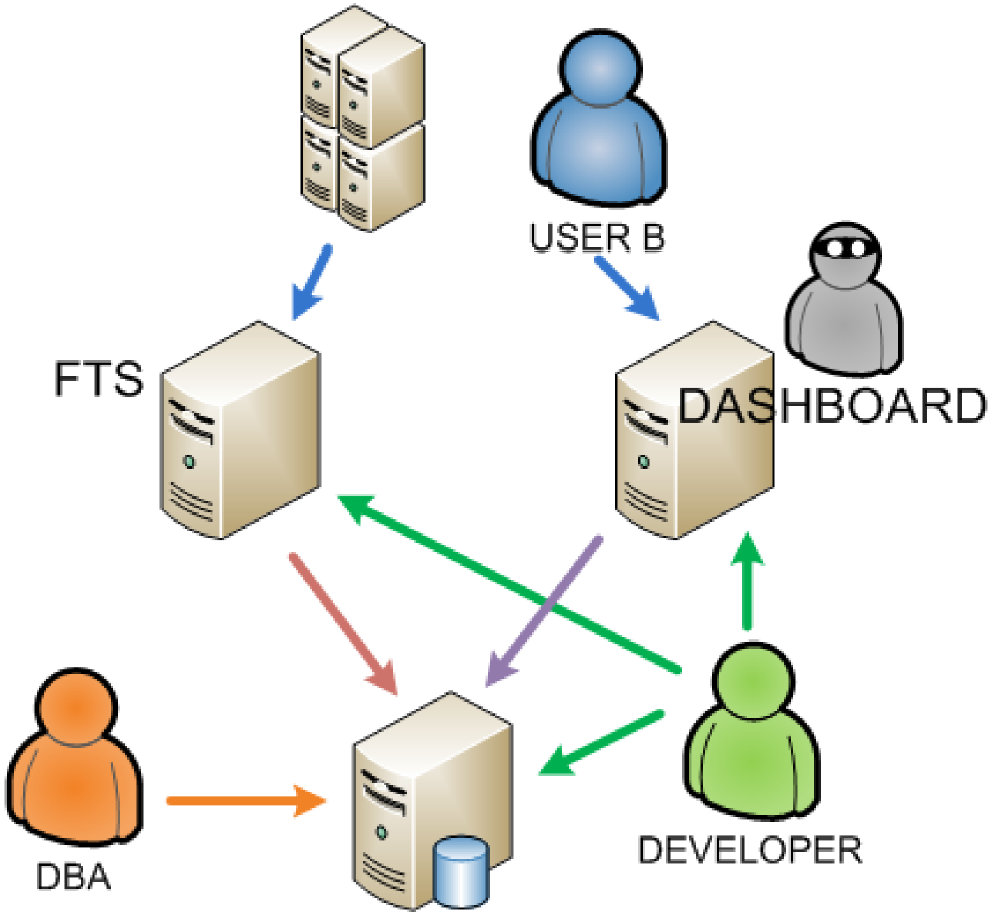
WINDOW = SELECT

IN DOOR = INSERT

OUT DOOR = DELETE

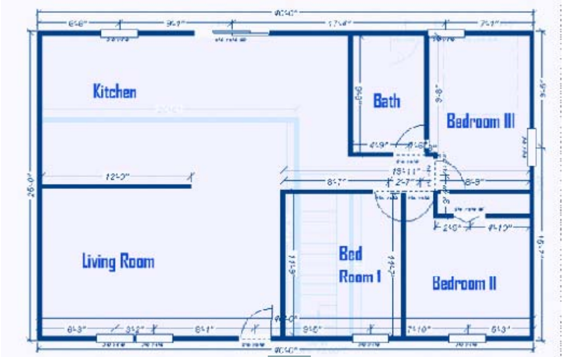
DOUBLE DOOR = UPDATE

3-Tier Application

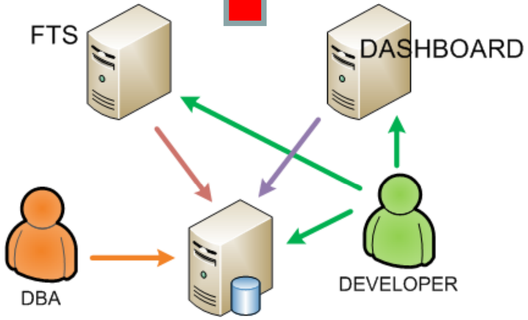


Abloy + floor plan + 3-tier app

+



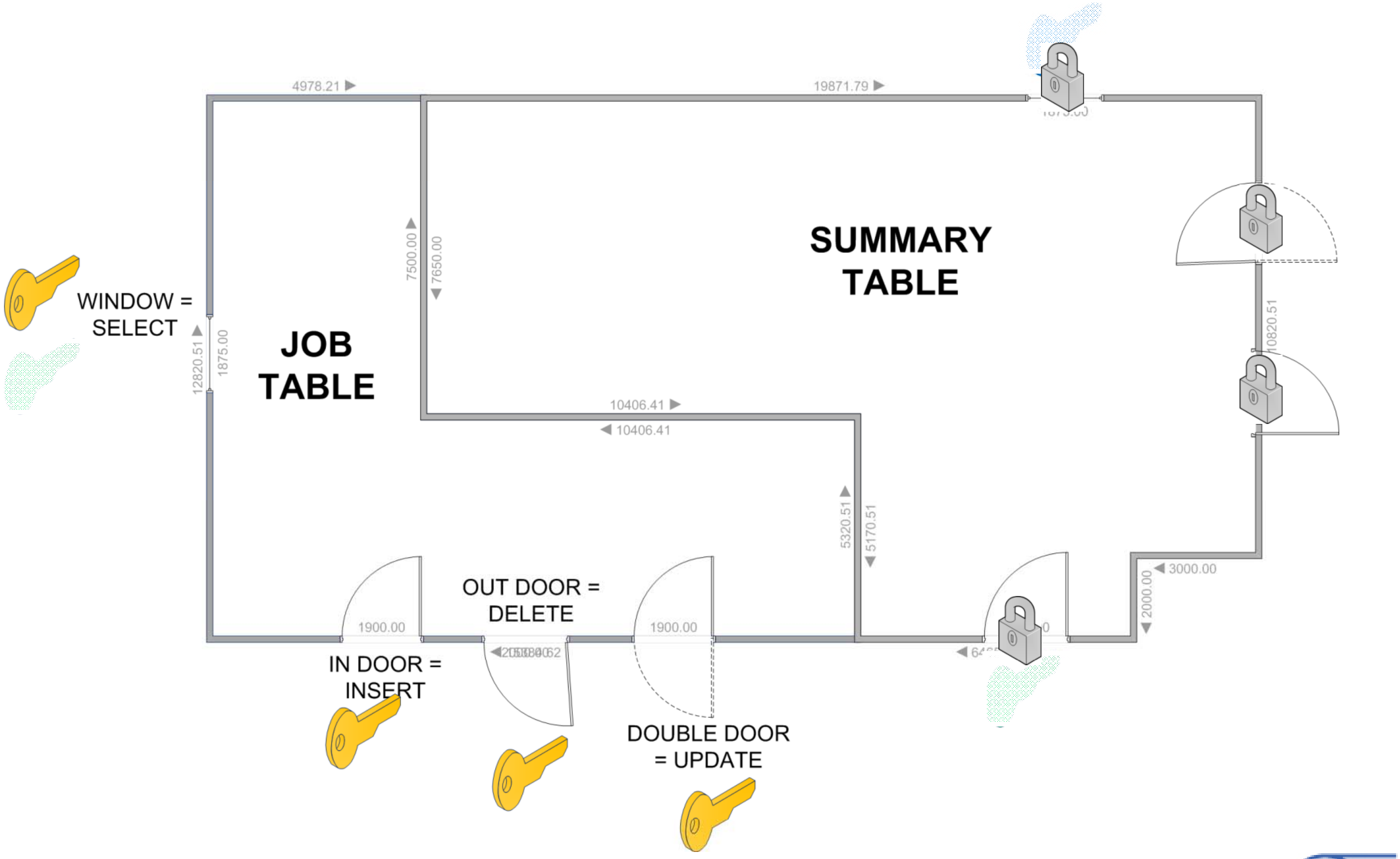
+



=

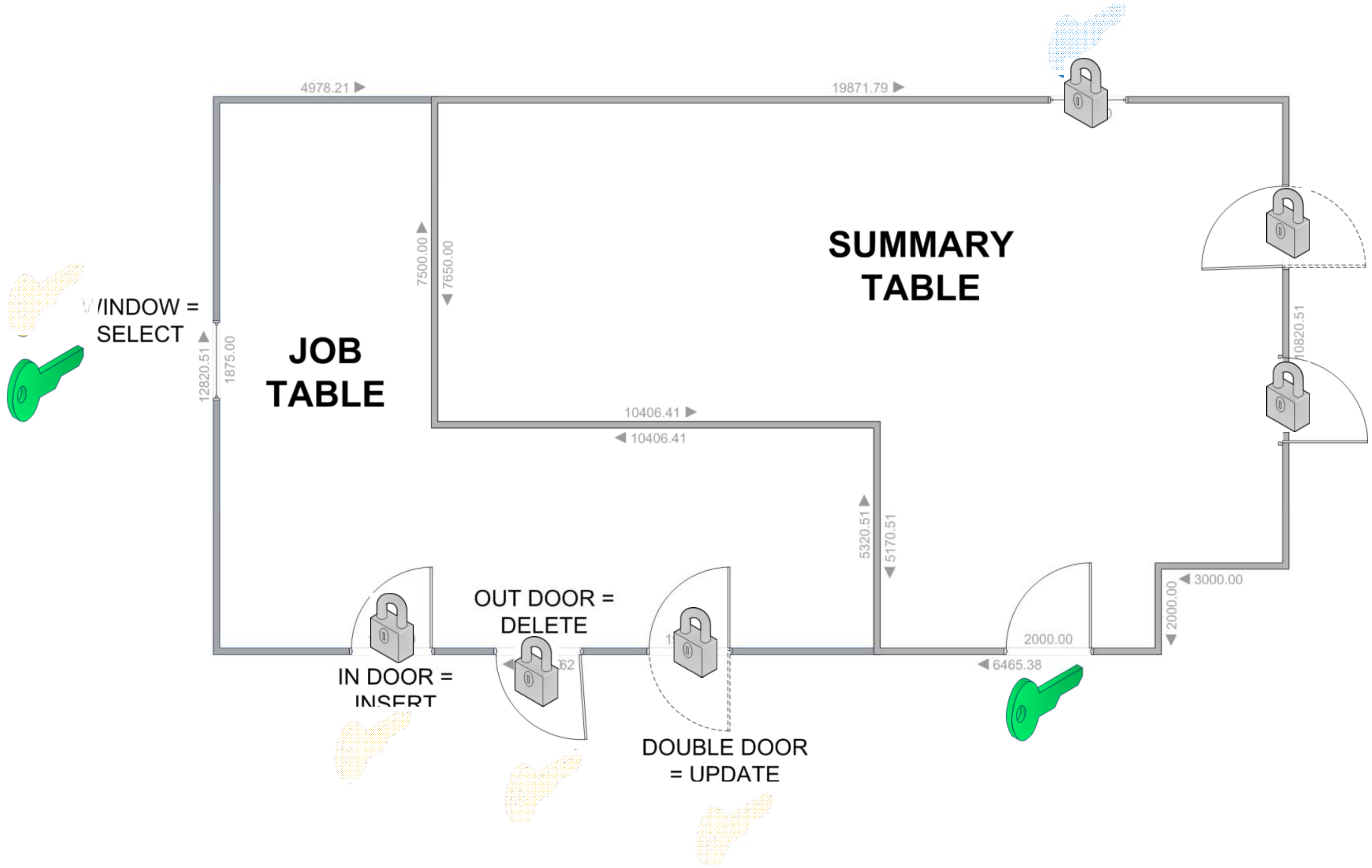
FTS abloy key

- FTS application access



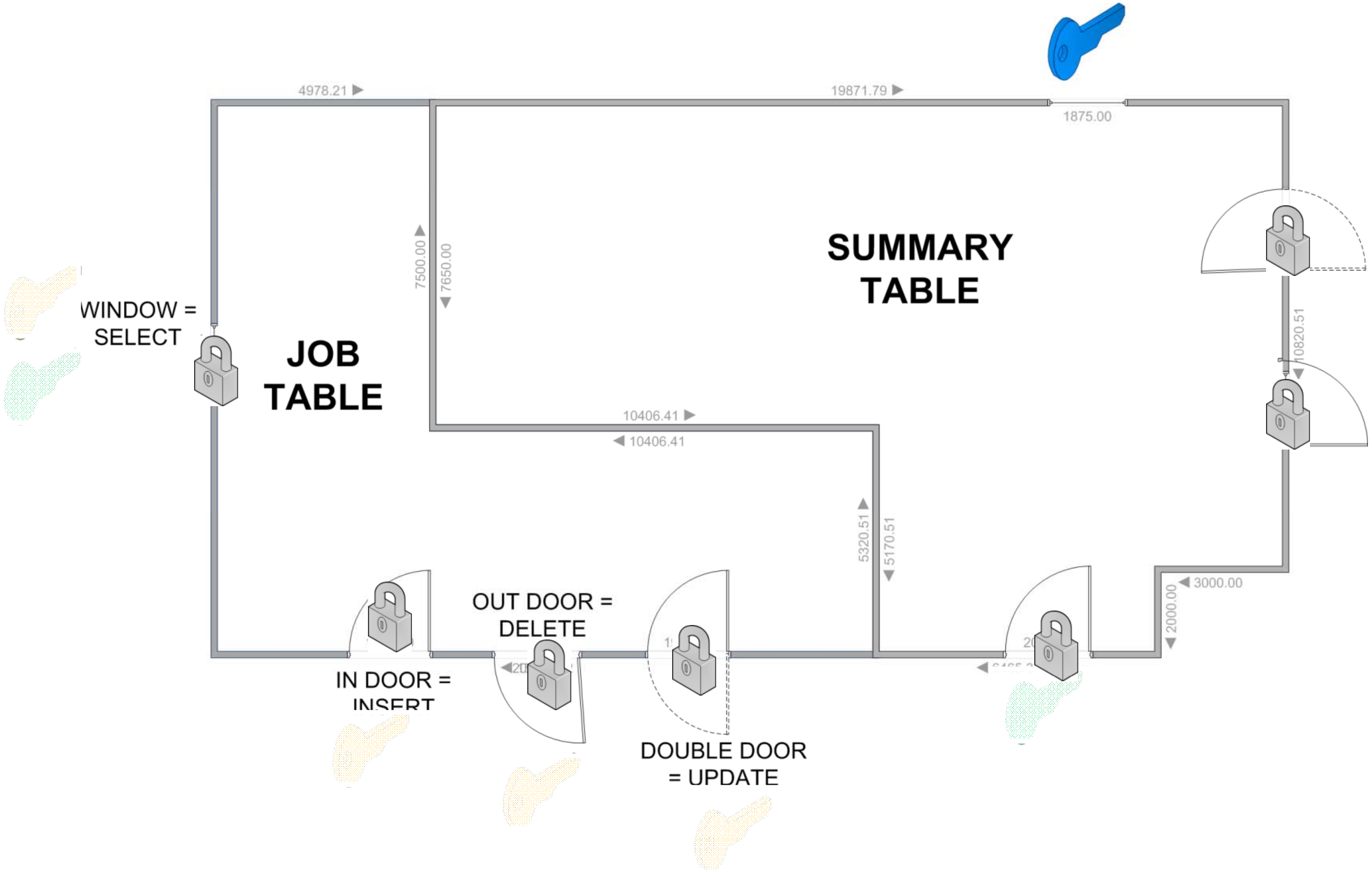
Dashboard abloy key 1

- Dashboard Writer application access



Dashboard abloy key 2

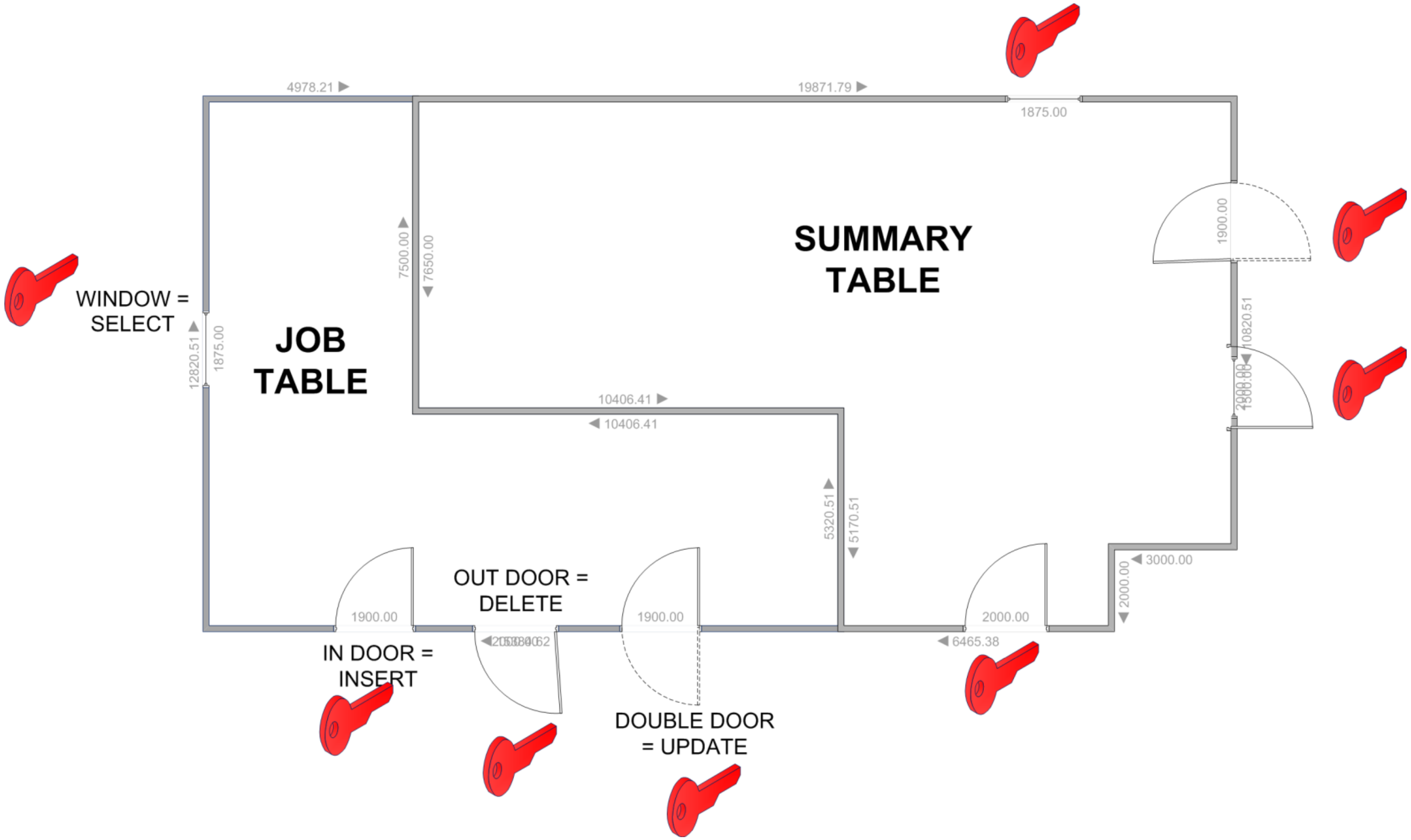
- Dashboard Reader application access



CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Developer abloy master key

- Developer

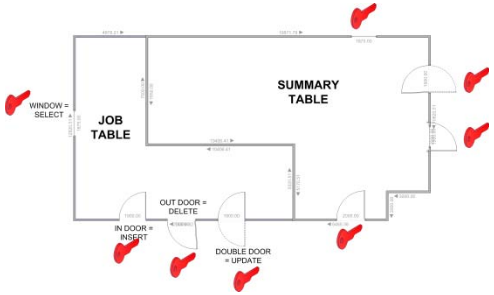


Application deployment

"I would like to put *FTS Dashboard* in production"

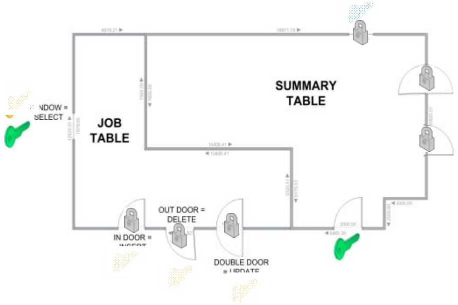
"Here are 3 abloy keys, one master and two configurable"

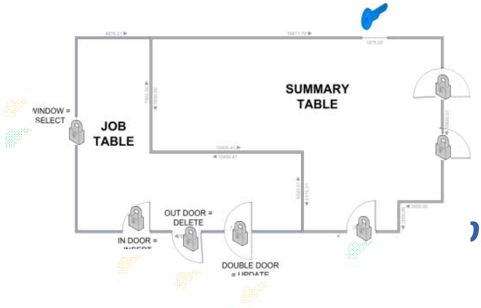
```
CREATE TABLE JOB;
CREATE TABLE SUMMARY;
```



- "Lets configure the other keys"

```
GRANT SELECT ON JOB TO DASHBOARD_W;
GRANT INSERT ON SUMMARY TO DASHBOARD_W;
GRANT SELECT ON SUMMARY TO DASHBOARD_R;
```





Application deployment

- Necessary to tell who is room owner:

```
SELECT FROM JOB;
```

  – ORA-00942: table or view does not exist

- `SELECT FROM DASHBOARD.JOB;`

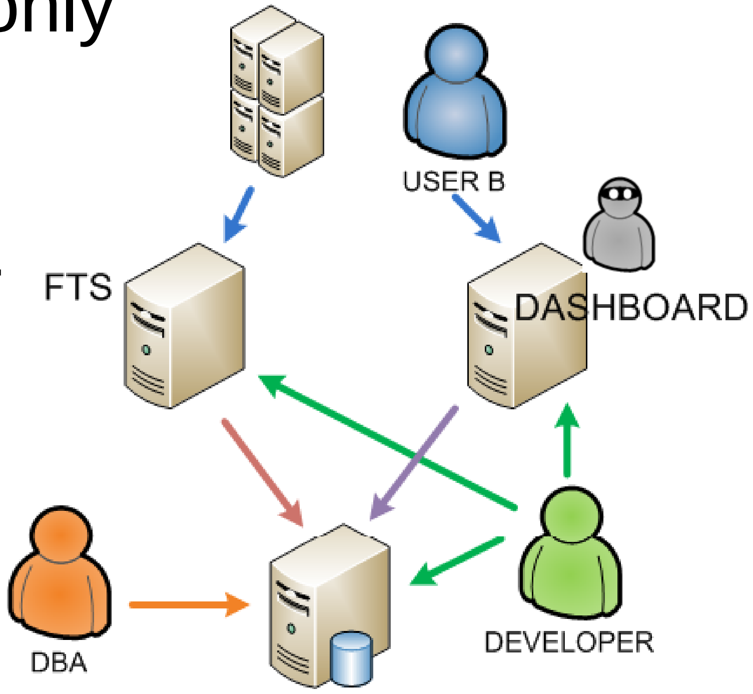
- Possibility to create a mapping:

```
CREATE SYNONYM JOB FOR DASHBOARD.JOB;
SELECT FROM JOB; (→SELECT FROM DASHBOARD.JOB)
```

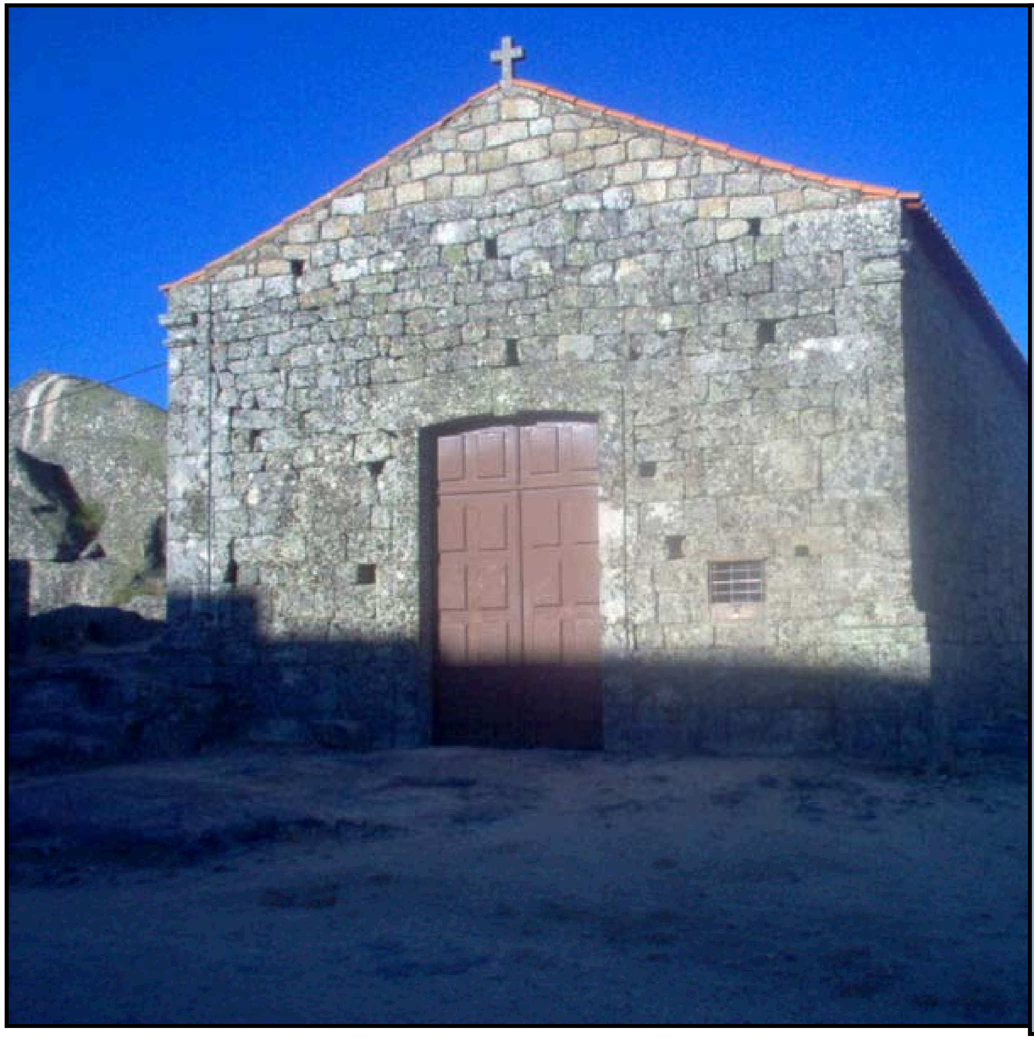
- Oracle bugs, might go to different house with same room name…

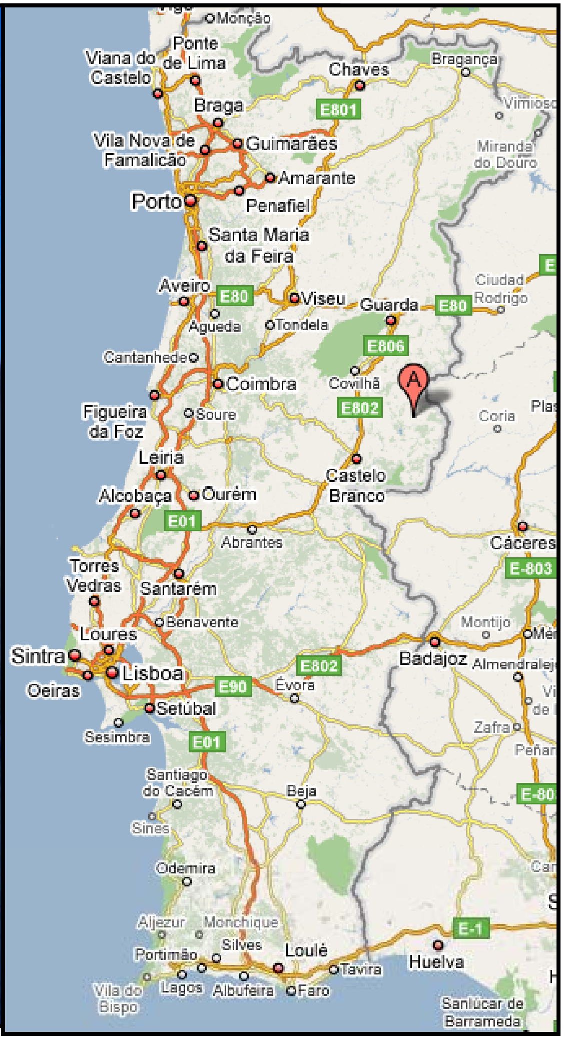
Application deployment - summary

- Each application should use different key configuration
- Each key should have only minimum privileges
- Necessary to tell owner name (**`dashboard.job`**)
- Endings _R/_W are just proposals

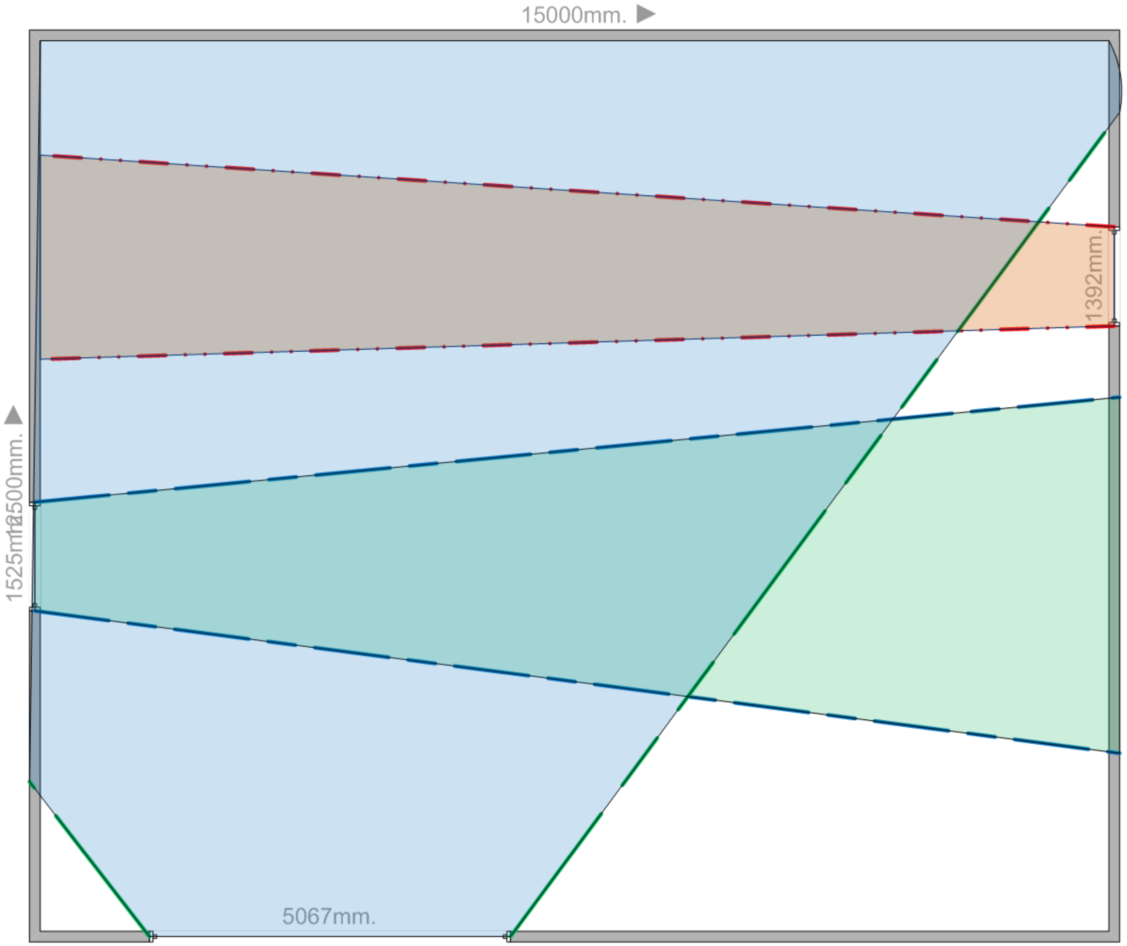


Updatable View





Updatable View



WINDOW 1 =
VIEW Alice

WINDOW 1 =
VIEW LHCb

WINDOW 1 =
VIEW ATLAS

Updatable View - example

- Window is filtered representation of a room
- Different VOs → different access rights
- Tables structure all the same

Build the window with right filters:

```
CREATE VIEW LHCB_ACL
 as SELECT … FROM ACL
 WHERE VO=‘LHCB’
 WITH CHECK OPTION;
```

- Add privilege to application key *(application is LFC for LHCB – username VOMS_LHCB):*

```
GRANT SELECT, INSERT on LHCB_ACL to LFC_LHCB;
```

Updatable View - example

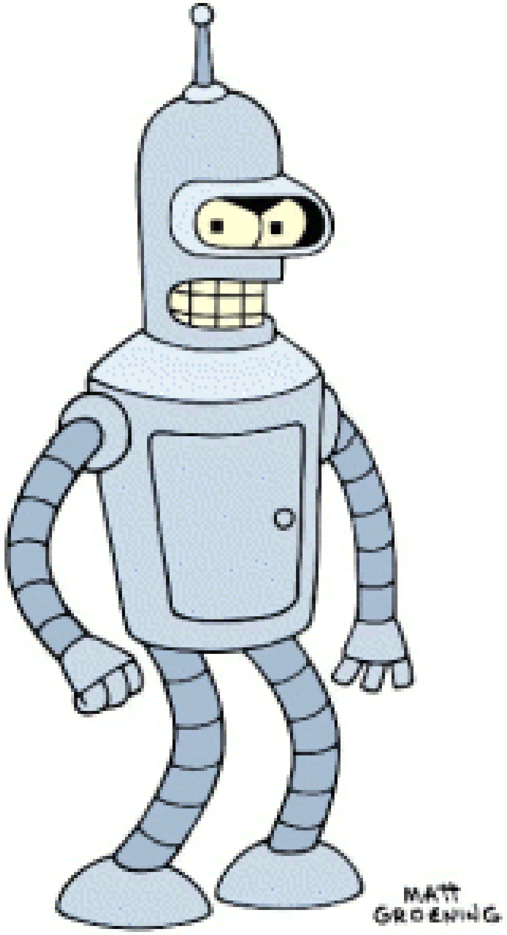
```
INSERT INTO LHCB_ACL(access,vo) VALUES ('w','CMS');
 ORA-01402: view WITH CHECK OPTION where-clause violation
```

```
INSERT INTO ACL(access,vo) VALUES ('w','CMS');
COMMIT;
```

```
SELECT * FROM LHCB_ACL;
no rows selected
```

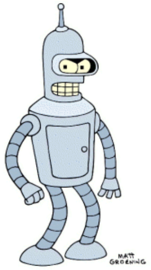
```
SELECT * FROM LHCB_ACL;
ORA-00942: table or view does not exist
```

Stored Procedures



Stored Procedures

- Robot does only programmed operation
- Can perform complex operations
- Privileges set who can call him
- Based on PL/SQL procedural language

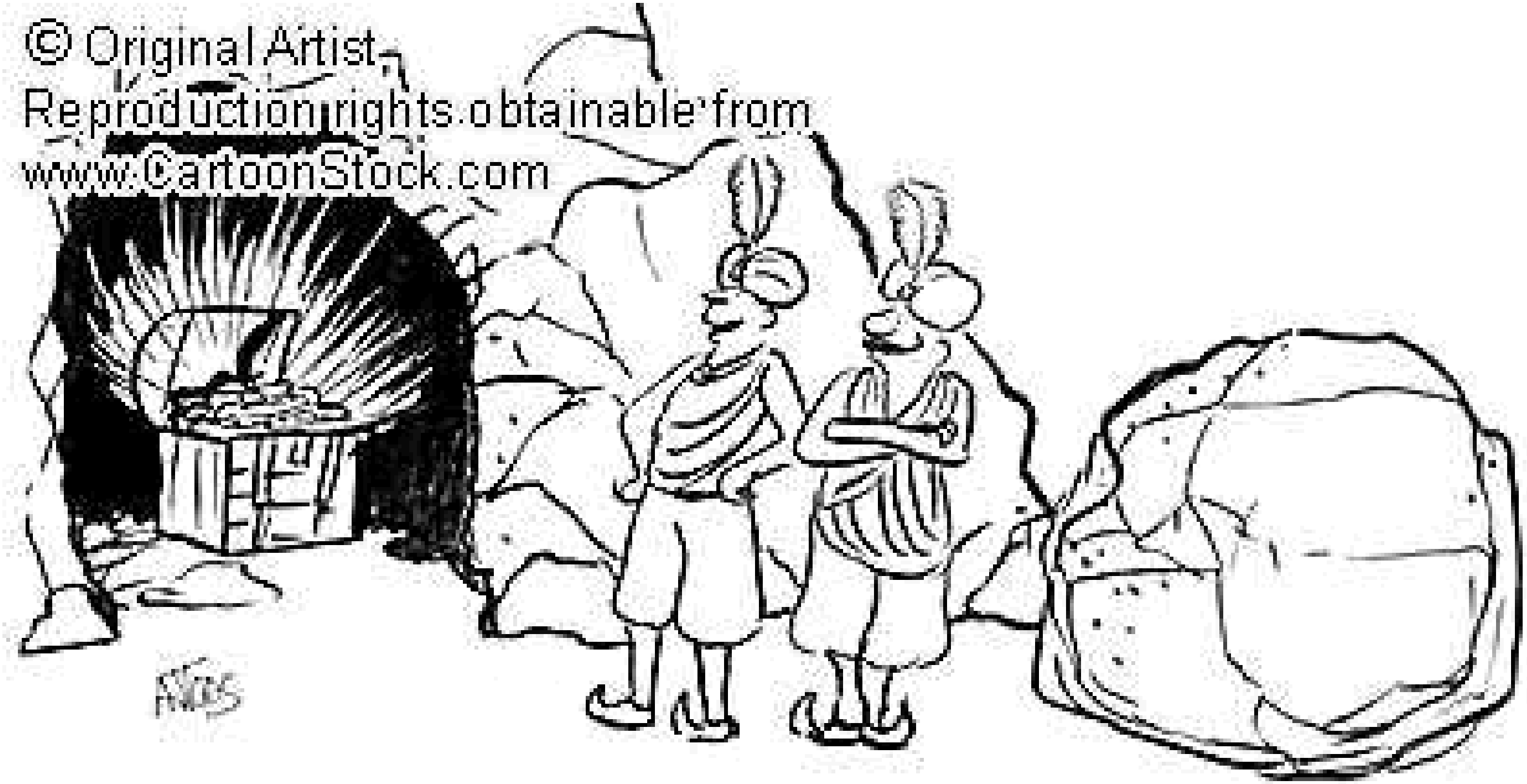


```
create procedure summarize is
    begin
    -- select some jobs
    -- if something do something else
    -- insert average on summarize just if something;
    end;
```

- `grant execute on summarize to dashboard_w;`

```
exec dashboard.summarize;
```

Role based access



© Original Artist
Reproduction rights obtainable from
www.CartoonStock.com

"I said 'Open Sesame' about forty times, and then I just said 'organic cauliflower'."

Role based access

- Hidden doors with ‘open sesame’
- Special privileges password protected
- Allows to group privileges

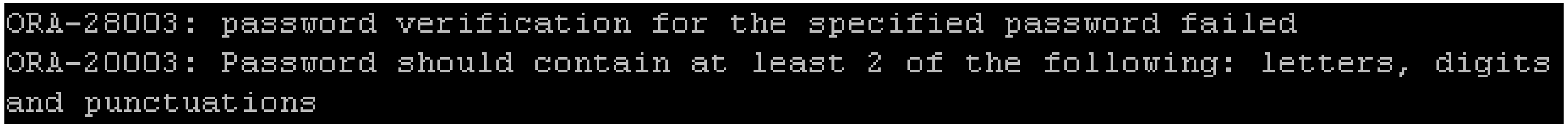
```
create role writer_role identified by x13y;
grant insert on summary to writer_role;
grant writer_role to dashboard_w;

insert into summary values (…);
 ORA-00942: table or view does not exist
set role writer_role identified by x13y;
insert into summary values (…);
 1 row created.
```

Owner vs Application accounts

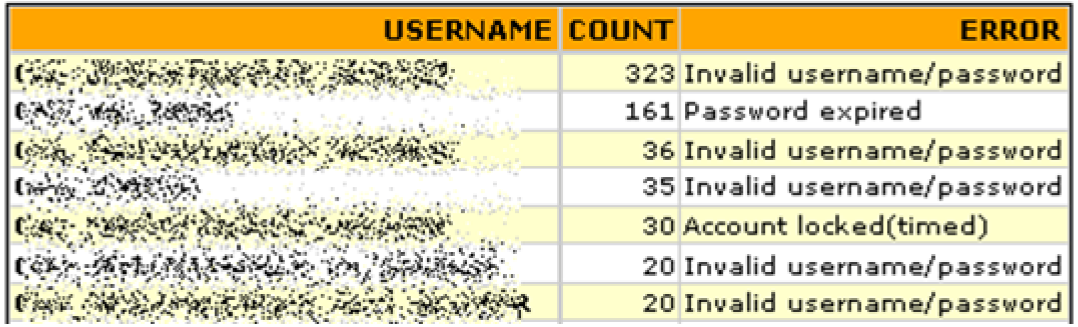
- The abloy keys are, in reality, Oracle accounts with different roles and profiles.
  - Owner – to be used by the developer
    - Master key
    - Can create objects (tables, views, sequences) and PL/SQL (functions, procedures, packages)
    - Responsible to configure application accounts
    - Maximum 10 simultaneous connections
    - Password expires after 1 year
  - Application accounts
    - No initial privileges (_W/_R are suggestions)
    - Max 400 sessions per DB instance (variable)
    - No password expiration (but recommended to change)
    - Can create synonyms (not recommended)

  - https://twiki.cern.ch/twiki/bin/view/PSSGroup/UserAccounts

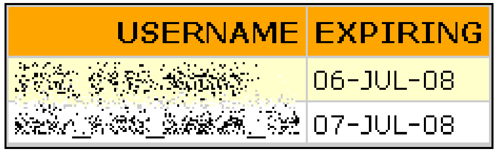
Profile - Password



```
ORA-28003: password verification for the specified password failed
ORA-20003: Password should contain at least 2 of the following: letters, digits
and punctuations
```



USERNAME	COUNT	ERROR
[redacted]	323	Invalid username/password
[redacted]	161	Password expired
[redacted]	36	Invalid username/password
[redacted]	35	Invalid username/password
[redacted]	30	Account locked(timed)
[redacted]	20	Invalid username/password
[redacted]	20	Invalid username/password



USERNAME	EXPIRING
[redacted]	06-JUL-08
[redacted]	07-JUL-08

- Prevention of brute force attacks
  - 1 minute locking after 5 failed attempts
- Expiration after 1 year
  - ☹ No email warning
- Cannot reuse password
- Follows CERN security recommendations

Summary

- Separate applications = separate privileges
- Give only minimum set of privileges
- Enhanced security with:
  - updatable views with check option
  - PL/SQL procedures
  - Password protected roles
- Avoid use of synonyms → use FQN
- Use different credentials in test/development
- Request necessary application accounts to your DBA
- Check if you gave too many privileges
  - Use `USER_TAB_PRIVS` view

Thanks!

Questions?