

## Chapter 11

# How to create views

Week 12, Week 13

# Objectives

## Applied

- Create and use views, including read-only and updatable views.
- Use Oracle SQL Developer to work with a view.

## Knowledge

- Describe a view.
- Describe the benefits of using views.
- Given a view, determine whether it is updatable.
- Describe the effect of the WITH CHECK OPTION clause on an updatable view.

# Knowledge Points in this lecture

- --1. Create view basics
- --2. Create views using subquery
- --3. Create or replace a view
- --4. Create views with calculated column in related query
- --5. Create a view using summary queries
- --6. FORCE keyword
- --7. Create updatable view
- --8. Create a read only view
- --9. Create a view using WITH CHECK OPTION
- --10. View update and base table constraints
- --11. Drop a view

## A CREATE VIEW statement

```
CREATE VIEW vendors_min AS
  SELECT vendor_name, vendor_state, vendor_phone
  FROM vendors;
```

## The **virtual table** that's represented by the view

	VENDOR_NAME	VENDOR_STATE	VENDOR_PHONE
1	US Postal Service	WI	(800) 555-1205
2	National Information Data Ctr	DC	(301) 555-8950
3	Register of Copyrights	DC	NULL
4	Jobtrak	CA	(800) 555-8725
5	Newbrige Book Clubs	NJ	(800) 555-9980

(122 rows)

Virtual table – a table whose data is not physically stored in the database.

## A statement that uses the Vendors\_Min view

```
SELECT * FROM vendors_min  
WHERE vendor_state = 'CA'  
ORDER BY vendor_name
```

## The result set that's returned

	VENDOR_NAME	VENDOR_STATE	VENDOR_PHONE
1	ASC Signs	CA	NULL
2	Abbey Office Furnishings	CA	(559) 555-8300
3	American Express	CA	(800) 555-3344
4	Aztek Label	CA	(714) 555-9000
5	BFI Industries	CA	(559) 555-1551

(75 rows)

## An UPDATE statement that uses a view to update the base table

```
UPDATE vendors_min  
SET vendor_phone = '(800) 555-3941'  
WHERE vendor_name = 'Register of Copyrights'
```

## The response from the system

```
1 rows updated
```

## A statement that drops a view

```
DROP VIEW vendors_min
```

## Some of the benefits provided by views

- Design independence
  - Add/Change view definition when base table is changed
- Data security
  - Restrict data access to only those visible in the view
- Simplified queries – query from views
- Updatability
  - Limited ability to update (insert, delete, update) data in the base table through views



# The syntax of the CREATE VIEW statement

```
CREATE [OR REPLACE] [{FORCE|NOFORCE}] VIEW view_name
    [(column_alias_1[, column_alias_2]...)]
AS
    select_statement
    [WITH {READ ONLY|CHECK OPTION}
        [CONSTRAINT constraint_name]]
```

## A statement that creates a view of vendors

```
CREATE VIEW vendors_phone_list AS
    SELECT vendor_name, vendor_contact_last_name,
           vendor_contact_first_name, vendor_phone
    FROM vendors
    WHERE vendor_id IN (SELECT vendor_id FROM invoices)
```

FORCE – create a view even when base table does not exist

Default: NOFORCE

WITH READ ONLY – view is read-only and cannot be updated

WITH CHECK OPTION – cannot modify a view such that modified data is no longer included in the view

## A CREATE VIEW statement that uses a join

```
CREATE OR REPLACE VIEW vendor_invoices AS
  SELECT vendor_name, invoice_number, invoice_date,
         invoice_total
  FROM vendors
  JOIN invoices
    ON vendors.vendor_id = invoices.vendor_id
```

## A CREATE VIEW statement that uses a subquery

```
CREATE OR REPLACE VIEW top5_invoice_totals AS
  SELECT vendor_id, invoice_total
  FROM (SELECT vendor_id, invoice_total FROM invoices
        ORDER BY invoice_total DESC)
 WHERE ROWNUM <= 5
```

## A statement that names all the view columns in its CREATE VIEW clause

```
CREATE OR REPLACE VIEW invoices_outstanding
    (invoice_number, invoice_date, invoice_total,
     balance_due)
AS
    SELECT invoice_number, invoice_date, invoice_total,
           invoice_total - payment_total - credit_total
    FROM invoices
    WHERE invoice_total - payment_total - credit_total > 0
```

If naming 1 column in CREATE VIEW statement,  
then must name all columns

## A statement that names just the calculated column in its SELECT clause

```
CREATE OR REPLACE VIEW invoices_outstanding AS
  SELECT invoice_number, invoice_date, invoice_total,
         invoice_total - payment_total - credit_total
         AS balance_due
  FROM invoices
 WHERE invoice_total - payment_total - credit_total > 0
```

## A CREATE VIEW statement that summarizes invoices by vendor

```
CREATE OR REPLACE VIEW invoice_summary AS
  SELECT vendor_name,
         COUNT(*) AS invoice_count,
         SUM(invoice_total) AS invoice_total_sum
  FROM vendors
       JOIN invoices
         ON vendors.vendor_id = invoices.vendor_id
 GROUP BY vendor_name
```

## A CREATE VIEW statement that uses FORCE

```
CREATE FORCE VIEW products_list AS  
    SELECT product_description, product_price  
    FROM products
```

### If you use the FORCE keyword...

- The view is created even if the base tables don't exist.

## Requirements for creating updatable views

- The select list can't include a DISTINCT clause.
- The select list can't include an aggregate function.
- The SELECT statement can't include a GROUP BY or HAVING clause.
- The view can't include the UNION operator.



## A CREATE VIEW statement that creates an **updatable view**

```
CREATE OR REPLACE VIEW balance_due_view AS
  SELECT vendor_name, invoice_number,
         invoice_total, payment_total, credit_total,
         invoice_total - payment_total - credit_total
         AS balance_due
  FROM vendors JOIN invoices
    ON vendors.vendor_id = invoices.vendor_id
 WHERE invoice_total - payment_total - credit_total > 0
```

## An UPDATE statement that uses the view

```
UPDATE balance_due_view
SET credit_total = 300
WHERE invoice_number = '989319-497'
```

## The response from the system

```
1 rows updated
```

## An UPDATE statement that attempts to use the view to update a calculated column

```
UPDATE balance_due_view  
SET balance_due = 0  
WHERE invoice_number = '989319-497';
```

## The response from the system

```
SQL Error: ORA-01733: virtual column not allowed here
```

## A CREATE VIEW statement that creates a read-only view

```
CREATE OR REPLACE VIEW balance_due_view AS
  SELECT vendor_name, invoice_number,
         invoice_total, payment_total, credit_total,
         invoice_total - payment_total - credit_total
         AS balance_due
  FROM vendors
  JOIN invoices
    ON vendors.vendor_id = invoices.vendor_id
 WHERE invoice_total - payment_total - credit_total > 0
WITH READ ONLY;
```

## An updatable view that has a **WITH CHECK OPTION** clause

```
CREATE OR REPLACE VIEW vendor_payment AS
  SELECT vendor_name, invoice_number, invoice_date,
         payment_date, invoice_total, credit_total,
         payment_total
  FROM vendors
    JOIN invoices
      ON vendors.vendor_id = invoices.vendor_id
 WHERE invoice_total - payment_total - credit_total >= 0
WITH CHECK OPTION
```

### If you use **WITH CHECK OPTION**...

- An error will occur if you modify a row so it's no longer included in the view.

## A statement that displays a row from the view

```
SELECT * FROM vendor_payment  
WHERE invoice_number = 'P-0608'
```

## The result set

VENDOR_NAME	INVOICE_NUMBER	INVOICE_DATE	PAYMENT_DATE	INVOICE_TOTAL	CREDIT_TOTAL	PAYMENT_TOTAL
1 Malloy Lithographing Inc	P-0608	11-APR-08	(null)	20551.18	1200	0

## An UPDATE statement that updates the view

```
UPDATE vendor_payment
SET payment_total = 400.00,
    payment_date = '01-AUG-14'
WHERE invoice_number = 'P-0608'
```

## The response from the system

1 rows updated

## The row after the update

	VENDOR_NAME	INVOICE_NUMBER	INVOICE_DATE	PAYMENT_DATE	INVOICE_TOTAL	CREDIT_TOTAL	PAYMENT_TOTAL
1	Malloy Lithographing Inc	P-0608	11-APR-08	01-AUG-08	20551.18	1200	400

## A statement that tries to update the view

```
UPDATE vendor_payment  
SET payment_total = 30000.00,  
    payment_date = '01-AUG-14'  
WHERE invoice_number = 'P-0608';
```

## The response from the system

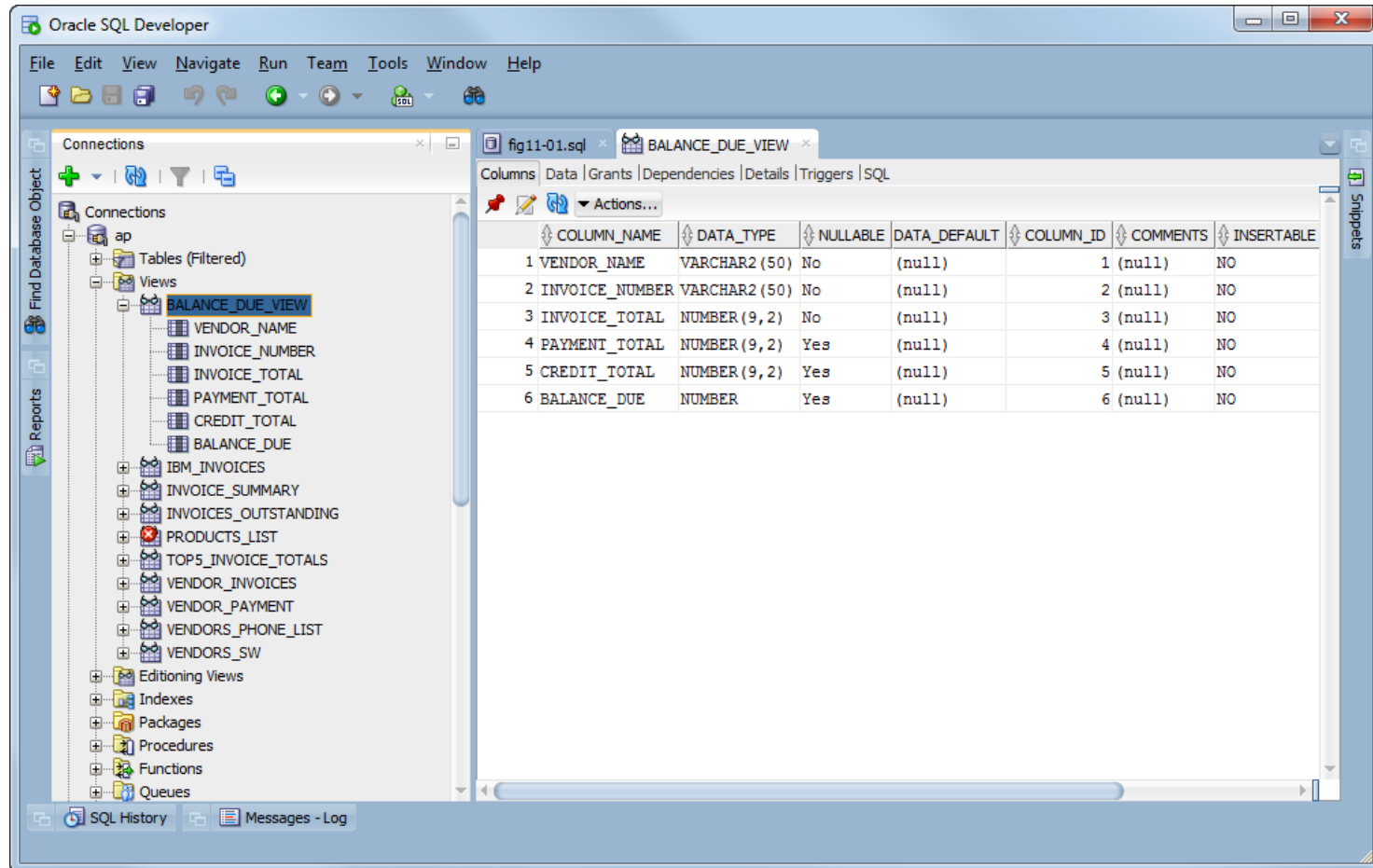
SQL Error: ORA-01402:

**view WITH CHECK OPTION where-clause violation**

The update would make the invoice 'P-0608' not satisfy the where condition in the view definition:

**invoice\_total - payment\_total - credit\_total >= 0**

# SQL Developer with the views for the AP schema





## A statement that creates an updatable view

```
CREATE OR REPLACE VIEW ibm_invoices AS  
  SELECT invoice_number, invoice_date, invoice_total  
  FROM invoices  
  WHERE vendor_id = 34;
```

## The contents of the view

	INVOICE_NUMBER	INVOICE_DATE	INVOICE_TOTAL
1	QP58872	25-FEB-08	116.54
2	Q545443	14-MAR-08	1083.58

## An INSERT statement that fails due to columns with null values

```
INSERT INTO ibm_invoices
  (invoice_number, invoice_date, invoice_total)
VALUES
  ('RA23988', '31-JUL-14', 417.34)
```

## The response from the system

```
SQL Error: ORA-01400:
cannot insert NULL into ("AP"."INVOICES"."INVOICE_ID")
```

## A DELETE statement that fails due to a foreign key constraint

```
DELETE FROM ibm_invoices  
WHERE invoice_number = 'Q545443'
```

## The response from the system

```
SQL Error: ORA-02292: integrity constraint  
(AP.LINE_ITEMS_FK_INVOICES) violated - child record found
```

## Two DELETE statements that succeed

```
DELETE FROM invoice_line_items
WHERE invoice_id = (SELECT invoice_id FROM invoices
                    WHERE invoice_number = 'Q545443');
```

```
DELETE FROM ibm_invoices
WHERE invoice_number = 'Q545443';
```

## The response from the system

```
1 rows deleted
```

## A statement that creates a view

```
CREATE VIEW vendors_sw AS  
SELECT *  
FROM vendors  
WHERE vendor_state IN ('CA', 'AZ', 'NV', 'NM')
```

## A statement that replaces the view with a new read-only view

```
CREATE OR REPLACE VIEW vendors_sw AS  
SELECT *  
FROM vendors  
WHERE vendor_state IN ('CA', 'AZ', 'NV', 'NM', 'UT', 'CO')  
WITH READ ONLY;
```

## A statement that drops the view

```
DROP VIEW vendors_sw
```