

# Chapter 12

# How to create views

# Objectives

## Applied

1. Create and use views, including read-only and updatable views.

## Knowledge

1. Describe a view.
2. Describe the benefits of using views.
3. Given a view, determine whether it is updatable.
4. Describe the effect of the WITH CHECK OPTION clause on an updatable 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
▶	US Postal Service	WI	(800) 555-1205
	National Information Data Ctr	DC	(301) 555-8950
	Register of Copyrights	DC	NULL
	Jobtrak	CA	(800) 555-8725
	Newbrige Book Clubs	NJ	(800) 555-9980

(122 rows)

## A SELECT statement that uses the view

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

## The result set returned by the SELECT statement

	vendor_name	vendor_state	vendor_phone
▶	Abbey Office Furnishings	CA	(559) 555-8300
	American Express	CA	(800) 555-3344
	ASC Signs	CA	NULL
	Aztek Label	CA	(714) 555-9000
	Bertelsmann Industry Svcs. Inc	CA	(805) 555-0584
	BFI Industries	CA	(559) 555-1551

(75 rows)

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

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

## A statement that drops the view

```
DROP VIEW vendors_min
```

## Some of the benefits provided by views

- Design independence
- Data security
- Simplified queries
- Updatability

## The syntax of the CREATE VIEW statement

```
CREATE [OR REPLACE] VIEW view_name
  [(column_alias_1[, column_alias_2]...)]
AS
  select_statement
  [WITH CHECK OPTION]
```

## A view of vendors that have invoices

```
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 DISTINCT vendor_id FROM invoices)
```

## A view 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 view that uses a LIMIT clause

```
CREATE OR REPLACE VIEW top5_invoice_totals AS
  SELECT vendor_id, invoice_total
  FROM invoices
  ORDER BY invoice_total DESC
  LIMIT 5
```

## A view that names all of its columns in the 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
```

## A view 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 view 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
```

## Requirements for creating updatable views

- The select list can't include a DISTINCT clause.
- The select list can't include aggregate functions.
- 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 = '9982771'
```

## The response from the system

(1 row affected)

## 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 = '9982771'
```

## The response from the system

```
Error Code: 1348. Column 'balance_due' is not updatable
```

## 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 try to modify a row so it's no longer included in the view.



## A SELECT 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
▶	Malloy Lithographing Inc	P-0608	2018-07-23	NULL	20551.18	1200.00	0.00

## An UPDATE statement that updates the view

```
UPDATE vendor_payment
SET payment_total = 400.00,
    payment_date = '2018-08-01'
WHERE invoice_number = 'P-0608'
```

## The response from the system

(1 row affected)

## The same row data after the update

	vendor_name	invoice_number	invoice_date	payment_date	invoice_total	credit_total	payment_total
▶	Malloy Lithographing Inc	P-0608	2018-07-23	2018-08-01	20551.18	1200.00	400.00

## An UPDATE statement that attempts to update the view

```
UPDATE vendor_payment
SET payment_total = 30000.00,
    payment_date = '2018-08-01'
WHERE invoice_number = 'P-0608';
```

## The response from the system

```
Error Code: 1369. CHECK OPTION failed 'ap.vendor_payment'
```

## 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
▶	QP58872	2018-05-07	116.54
	Q545443	2018-06-09	1083.58

## An INSERT statement that fails due to columns that don't have values

```
INSERT INTO ibm_invoices
  (invoice_number, invoice_date, invoice_total)
VALUES
  ('RA23988', '2018-07-31', 417.34)
```

## The response from the system

```
Error Code: 1423. Field of view 'ap.ibm_invoices'
underlying table doesn't have a default value
```

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

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

## The response from the system

```
Error Code: 1451. Cannot delete or update a parent row: a  
foreign key constraint fails ('ap'. 'invoice_line_items',  
CONSTRAINT 'line_items_fk_invoices' FOREIGN KEY  
( 'invoice_id') REFERENCES 'invoices' ('invoice_id'))
```

## 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 row affected)

## 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 view

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

## A statement that drops the view

```
DROP VIEW vendors_sw
```