CREATE SCHEMA video\_games;

USE video\_games;

CREATE TABLE video\_game\_sales (

`title` VARCHAR(150),

`console` VARCHAR(10),

`genre` VARCHAR(20),

`publisher` VARCHAR(50),

`developer` VARCHAR(70),

`critic\_score` FLOAT,

`total\_sales` FLOAT,

`na\_sales` FLOAT,

`jp\_sales` FLOAT,

`pal\_sales` FLOAT,

`other\_sales` FLOAT,

`release\_date` VARCHAR(15),

`last\_update` VARCHAR(15)

);

SELECT

\*

FROM

video\_game\_sales

LIMIT 5;

-- Create a copy of dataset to use for analysis

CREATE TABLE sales\_copy LIKE video\_game\_sales;

INSERT sales\_copy

SELECT \*

FROM video\_game\_sales;

-- Data Cleaning

WITH duplicate\_cte AS (

SELECT \*,

ROW\_NUMBER() OVER(

PARTITION BY title, console, genre, publisher, developer, critic\_score, total\_sales,

na\_sales, jp\_sales, pal\_sales, other\_sales, release\_date, last\_update) AS row\_num

FROM sales\_copy)

SELECT \*

FROM duplicate\_cte

WHERE row\_num > 1;

CREATE TABLE `sales\_copy\_2` (

`title` VARCHAR(150) DEFAULT NULL,

`console` VARCHAR(10) DEFAULT NULL,

`genre` VARCHAR(20) DEFAULT NULL,

`publisher` VARCHAR(50) DEFAULT NULL,

`developer` VARCHAR(70) DEFAULT NULL,

`critic\_score` FLOAT DEFAULT NULL,

`total\_sales` FLOAT DEFAULT NULL,

`na\_sales` FLOAT DEFAULT NULL,

`jp\_sales` FLOAT DEFAULT NULL,

`pal\_sales` FLOAT DEFAULT NULL,

`other\_sales` FLOAT DEFAULT NULL,

`release\_date` VARCHAR(15) DEFAULT NULL,

`last\_update` VARCHAR(15) DEFAULT NULL,

row\_num INT

) ENGINE=INNODB DEFAULT CHARSET=UTF8MB4 COLLATE = UTF8MB4\_0900\_AI\_CI;

INSERT INTO sales\_copy\_2

SELECT \*,

ROW\_NUMBER() OVER(

PARTITION BY title, console, genre, publisher, developer, critic\_score, total\_sales,

na\_sales, jp\_sales, pal\_sales, other\_sales, release\_date, last\_update) AS row\_num

FROM sales\_copy;

DELETE FROM sales\_copy\_2

WHERE

row\_num > 1;

SELECT

\*

FROM

sales\_copy\_2

WHERE

row\_num > 1;

-- Standardising Data

SELECT DISTINCT

title

FROM

sales\_copy\_2

ORDER BY title;

-- Some records have the console in the genre column and the genre in the publisher column.

-- Grouping consoles and genres with only a few records into other.

SELECT

\*

FROM

sales\_copy\_2

WHERE

genre IN ('3DS' , 'All',

'DC',

'DS',

'GB',

'GBA',

'GC',

'GEN',

'Linux',

'MSD',

'N64',

'NES',

'NS',

'OSX',

'PC',

'PCE',

'PS',

'PSV',

'PS2',

'PS3',

'PS4',

'PS5',

'PSP',

'DS',

'PSN',

'SAT',

'SNES',

'Wii',

'WiiU',

'WS',

'X360',

'XOne',

'XB',

'XBL',

'XS');

UPDATE sales\_copy\_2

SET

console = genre,

genre = publisher

WHERE

genre IN ('3DS' , 'All',

'DC',

'DS',

'GB',

'GBA',

'GC',

'GEN',

'Linux',

'MSD',

'N64',

'NES',

'NS',

'OSX',

'PC',

'PCE',

'PS',

'PSV',

'PS2',

'PS3',

'PS4',

'PS5',

'PSP',

'DS',

'PSN',

'SAT',

'SNES',

'Wii',

'WiiU',

'WS',

'X360',

'XOne',

'XB',

'XBL',

'XS');

CREATE TEMPORARY TABLE miscellaneous\_genres AS

SELECT genre

FROM sales\_copy\_2

GROUP BY genre

HAVING COUNT(\*) <= 10;

UPDATE sales\_copy\_2

SET

genre = 'Other'

WHERE

genre IN (SELECT

genre

FROM

miscellaneous\_genres);

SELECT

console, COUNT(\*)

FROM

sales\_copy\_2

GROUP BY console;

CREATE TEMPORARY TABLE miscellaneous\_consoles AS

SELECT console

FROM sales\_copy\_2

GROUP BY console

HAVING COUNT(\*) <= 75;

UPDATE sales\_copy\_2

SET

console = 'Other'

WHERE

console IN (SELECT

console

FROM

miscellaneous\_consoles);

SELECT

\*

FROM

sales\_copy\_2

WHERE

console = 'All';

SELECT

publisher, COUNT(\*)

FROM

sales\_copy\_2

GROUP BY publisher

HAVING COUNT(\*) > 15;

-- Changing records with publishers which are consoles or genres to unknown publisher.

CREATE TEMPORARY TABLE publisher\_genres (

SELECT publisher, COUNT(\*)

FROM sales\_copy\_2

GROUP BY publisher

HAVING publisher IN (SELECT DISTINCT genre FROM sales\_copy\_2));

SELECT

\*

FROM

sales\_copy\_2

WHERE

publisher IN (SELECT

publisher

FROM

publisher\_genres);

UPDATE sales\_copy\_2

SET

publisher = 'Unknown'

WHERE

publisher IN (SELECT

publisher

FROM

publisher\_genres);

CREATE TEMPORARY TABLE publisher\_consoles (

SELECT publisher, COUNT(\*)

FROM sales\_copy\_2

GROUP BY publisher

HAVING publisher IN (SELECT DISTINCT console FROM sales\_copy\_2));

SELECT

\*

FROM

sales\_copy\_2

WHERE

publisher IN (SELECT

publisher

FROM

publisher\_consoles);

UPDATE sales\_copy\_2

SET

publisher = 'Unknown'

WHERE

publisher IN (SELECT

publisher

FROM

publisher\_consoles);

SELECT

publisher, COUNT(\*)

FROM

sales\_copy\_2

GROUP BY publisher

ORDER BY publisher;

CREATE TEMPORARY TABLE other\_publishers (

SELECT publisher, COUNT(\*) AS count

FROM sales\_copy\_2

GROUP BY publisher);

UPDATE sales\_copy\_2

SET

publisher = 'Other'

WHERE

publisher IN (SELECT

publisher

FROM

other\_publishers

WHERE

count < 21);

SELECT

developer, COUNT(\*)

FROM

sales\_copy\_2

GROUP BY developer

HAVING COUNT(\*) > 20

ORDER BY developer;

CREATE TEMPORARY TABLE other\_developers (

SELECT developer, COUNT(\*) AS count

FROM sales\_copy\_2

GROUP BY developer

HAVING count < 21);

UPDATE sales\_copy\_2

SET

developer = 'Other'

WHERE

developer IN (SELECT

developer

FROM

other\_developers

WHERE

count < 21);

-- Critic score is supposed to be between 1 and 10

SELECT

critic\_score, COUNT(\*)

FROM

sales\_copy\_2

GROUP BY critic\_score

HAVING critic\_score < 1 OR critic\_score > 10

ORDER BY critic\_score;

UPDATE sales\_copy\_2

SET

critic\_score = NULL

WHERE

critic\_Score < 1 OR critic\_score > 10;

-- Finding titles with total sales that are different than the sum of their components.

SELECT DISTINCT

total\_sales,

ROUND(COALESCE(na\_sales, 0) + COALESCE(jp\_sales, 0) + COALESCE(pal\_sales, 0) + COALESCE(other\_sales, 0),

2)

FROM

sales\_copy\_2

WHERE

ROUND(COALESCE(na\_sales, 0) + COALESCE(jp\_sales, 0) + COALESCE(pal\_sales, 0) + COALESCE(other\_sales, 0),

2) != total\_sales;

UPDATE sales\_copy\_2

SET

total\_sales = ROUND(COALESCE(na\_sales, 0) + COALESCE(jp\_sales, 0) + COALESCE(pal\_sales, 0) + COALESCE(other\_sales, 0),

2)

WHERE

ROUND(COALESCE(na\_sales, 0) + COALESCE(jp\_sales, 0) + COALESCE(pal\_sales, 0) + COALESCE(other\_sales, 0),

2) != total\_sales;

SELECT DISTINCT

other\_sales

FROM

sales\_copy\_2

ORDER BY other\_sales;

SELECT

release\_date, COUNT(\*)

FROM

sales\_copy\_2

GROUP BY release\_date

HAVING release\_date NOT LIKE '%/%/%';

UPDATE sales\_copy\_2

SET

release\_date = NULL

WHERE

release\_date NOT LIKE '%/%/%';

UPDATE sales\_copy\_2

SET

last\_update = NULL

WHERE

last\_update NOT LIKE '%/%/%';

SELECT

STR\_TO\_DATE(release\_date, '%d/%m/%Y')

FROM

sales\_copy\_2;

UPDATE sales\_copy\_2

SET

release\_date = STR\_TO\_DATE(release\_date, '%d/%m/%Y');

UPDATE sales\_copy\_2

SET

last\_update = STR\_TO\_DATE(last\_update, '%d/%m/%Y');

SELECT

\*

FROM

sales\_copy\_2

LIMIT 5;

-- Missing Values

SELECT

title

FROM

sales\_copy\_2

WHERE

title IS NULL OR TRIM(title) = '';

SELECT

console

FROM

sales\_copy\_2

WHERE

console IS NULL OR TRIM(console) = '';

SELECT

genre

FROM

sales\_copy\_2

WHERE

genre IS NULL OR TRIM(genre) = '';

SELECT

publisher

FROM

sales\_copy\_2

WHERE

publisher IS NULL

OR TRIM(publisher) = '';

SELECT

developer

FROM

sales\_copy\_2

WHERE

developer IS NULL

OR TRIM(developer) = '';

SELECT

critic\_score

FROM

sales\_copy\_2

WHERE

critic\_score IS NULL

OR TRIM(critic\_score) = '';

SELECT

total\_sales

FROM

sales\_copy\_2

WHERE

total\_sales IS NULL

OR TRIM(total\_sales) = '';

SELECT

na\_sales

FROM

sales\_copy\_2

WHERE

na\_sales IS NULL OR TRIM(na\_sales) = '';

SELECT

jp\_sales

FROM

sales\_copy\_2

WHERE

jp\_sales IS NULL OR TRIM(jp\_sales) = '';

SELECT

pal\_sales

FROM

sales\_copy\_2

WHERE

pal\_sales IS NULL

OR TRIM(pal\_sales) = '';

SELECT

other\_sales

FROM

sales\_copy\_2

WHERE

other\_sales IS NULL

OR TRIM(other\_sales) = '';

SELECT

release\_date

FROM

sales\_copy\_2

WHERE

release\_date IS NULL

OR TRIM(release\_date) = '';

SELECT

release\_date, critic\_score

FROM

sales\_copy\_2

WHERE

release\_date IS NULL

OR TRIM(release\_date) = ''

AND critic\_score IS NULL;

-- All records with null release\_dates have null critic\_scores.

SELECT

\*

FROM

sales\_copy\_2

WHERE

last\_update IS NULL;

ALTER TABLE sales\_copy\_2

DROP COLUMN row\_num;