Question 1:

```
CREATE TABLE Publishers (
  publisher_id VARCHAR(100) PRIMARY KEY,
  location VARCHAR(255),
  contact_number VARCHAR(20)
);

CREATE TABLE Books (
  book_code INT PRIMARY KEY,
  book_title VARCHAR(255),
  publisher_id VARCHAR(100),
  FOREIGN KEY (publisher_id) REFERENCES Publishers(publisher_id)
);

CREATE TABLE Users (
  user_id INT PRIMARY KEY,
  full_name VARCHAR(100)
);

CREATE TABLE Rentals (
  book_code INT,
  user_id INT,
  rental_date DATE,
  PRIMARY KEY (book_code, user_id, rental_date),
  FOREIGN KEY (book_code) REFERENCES Books(book_code),
  FOREIGN KEY (user_id) REFERENCES Users(user_id)
);
```

a)

```
SELECT DISTINCT u.user_id, u.full_name
FROM Users u
JOIN Rentals r ON u.user_id = r.user_id
JOIN Books b ON r.book_code = b.book_code
WHERE b.publisher_id = 'books around the world’;
```

b)

```
SELECT u.user_id, u.full_name
FROM Users u
WHERE NOT EXISTS (
  SELECT 1
  FROM Books b
  WHERE b.publisher_id = ‘books around the world’
  AND NOT EXISTS (
    SELECT 1
    FROM Rentals r
    WHERE r.book_code = b.book_code AND r.user_id = u.user_id
  )
);
```

c)

```
SELECT b.publisher_id, u.user_id, u.full_name
FROM Users u
JOIN Rentals r ON u.user_id = r.user_id
JOIN Books b ON r.book_code = b.book_code
GROUP BY b.publisher_id, u.user_id, u.full_name
HAVING COUNT(*) > 5;
```

d)

```
SELECT
  (SELECT COUNT(*) FROM Rentals) * 1.0 / COUNT(*) AS avg_books_per_user
FROM Users;
```

Question 2 :

a)

```
SELECT   *,
  SUM(quantity) OVER (ORDER BY sale_date ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS running_total
FROM Sales;
```

b)

```
SELECT *
FROM (
  SELECT *,
        ROW_NUMBER() OVER (PARTITION BY product_code ORDER BY quantity ASC) AS rank_order
  FROM Sales
) ranked_sales
WHERE rank_order <= 2;
```